AFIT/GSO/ENS/99M-01

# MULTI-DISCIPLINE NETWORK VULNERABILITY ASSESSMENT

THESIS

Karilynne Wallace, Captain, USAF

AFIT/GSO/ENS/99M-01

## Disclaimer

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

AFIT/GSO/ENS/99M-01

# MULTI-DISCIPLINE NETWORK VULNERABILITY ASSESSMENT

## THESIS

Presented to the faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Space Operations

Karilynne Wallace, B.S.

Captain, USAF

March 1999

AFIT/GSO/ENS/99M-01

# MULTI-DISCIPLINE NETWORK VULNERABILITY ASSESSMENT

Karilynne Wallace, B.S.
Captain, USAF

Approved:

W. Paul Murdock, Jr. — 2 MAR 99

Advisor — date
W. Paul Murdock, Jr., Maj, USAF
Assistant Professor of Operations Research
Department of Operational Sciences

Stuart C. Kramer — 2 MAR 99

Reader — date
Stuart Kramer, Lt Col, USAF
Associate Professor of Systems Engineering
Department of Aeronautical and Astronautical Engineering

## Acknowledgements

I would like to acknowledge the many people without whom I could not have completed this thesis. First, I would like to thank my advisor, Major Murdock. His guidance and insight throughout the research process, as well as his editing during the preparation of this document, were invaluable. I would also like to thank my sponsor, Mr. Steve Pease from AFSPACECOM/DOIK, for the latitude provided to me in this endeavor. Next, I would like to thank my committee member, Lieutenant Colonel Kramer, for his timely support to my research. Most importantly, I would like to thank my husband Charles and my son Jaron. Their understanding, support, and love enabled me to make it through the long hours and still keep things in perspective. Thanks! I couldn't have done it without you.

Karilynne Wallace

# Table of Contents

Page

# List of Figures

Page

PAGE

# List of Tables

Page

## Abstract

"Air Force Space Command has a mission to provide real-time, survivable, and enduring communications, surveillance, environmental monitoring, navigation, and warning..." according to Air Force Space Handbook – A War Fighters Guide to Space. In order to prevent disruption of the critical networks in Space Command, a vulnerability assessment must be performed. The set of components called a vulnerability set, which could cause network disruption, must be identified and prioritized. In order to move beyond subjective prioritization, implementation of a quantitative methodology that measures the value of each vulnerability set is needed.

This thesis proposes a technique for vulnerability assessment of a multi-disciplined network consisting of components that satisfy primary functions and secondary supporting functions. A *Network Vulnerability Assessment Tool* was developed and used to analyze the network. The tool models the network as nodes and links. A cut-set algorithm generates potential vulnerability sets. The overall objective is to find sets that cause critical network disruptions. Value-focused thinking and decision analysis techniques are used to rank the vulnerability sets according to decision maker preferences. The value model is applied at the network and sub-network level. This allows for analysis of sub-network vulnerabilities that are not evident in the general network analysis. The network vulnerability sets are ranked to provide an overall vulnerability assessment. Sensitivity analysis provides insight into the driving factors of the vulnerability set prioritization.

MULTI-DISCIPLINE NETWORK VULNERABILITY ASSESSMENT

# I. Introduction

## Background

> *"The principles of war: mass, objective, surprise, maneuver, the offensive, simplicity, unity of command, economy of force, and security apply fully and completely to space operations. As we have moved into space, we have not found reason to question these principles, nor have we discovered new ones." [10:69]*

"The United States must win and maintain the capability to control space in order to assure the progress and pre-eminence of the free nations. If liberty and freedom are to remain in the world, the United States and its allies must be in position to control space" [9:19]. These words by General Thomas D. White, Air Force Chief of Staff, 1955, were restated in the Long Range Plan (LRP) of United States Space Command.

> *As space products and services become ever more interwoven with our nation's politics, economics, culture, and security, they become an increasingly lucrative target for potential adversaries. A future foe could gain an advantage by denying, disrupting, or destroying our ability to access and use space* [9:19].

As you can imagine, these words are just as critical today as they were in 1955.

Air Force Doctrine states it in a slightly different, but direct way. "Gaining air, and space superiority is a primary goal of a military campaign and must be achieved early to ensure freedom of action" [11:7].

It is important to discuss how space superiority can be achieved. To start the discussion, space control must be defined. Is the control of space really a critical part of the United States Air Force? If the US has control of space, has space superiority been achieved? If so, must the US take any further action to maintain this state? These questions are approached in the following quotation from Air Force Doctrine.

> *Space Control is the means by which space superiority is gained and maintained to assure friendly forces can use the space environment while denying its use to the enemy. To accomplish this, space forces must survey space, protect the ability to use space, prevent adversaries from exploiting US or allied space services, and negate the ability for adversaries to exploit their space forces [11:8].*

The statements above highlight the importance of maintaining space control. This control can be gained and maintained by counterspace activities, both offensive and defensive. Defensive activities can be both active and passive. This thesis focuses on the passive, defensive aspect of counterspace. "The objectives of passive defense are to reduce the vulnerabilities and to protect and increase the survivability of friendly space forces and the information they provide" [11:10].

It is very important to protect the nation's space systems to maintain space control. According to the LRP,

> *Protection requires five key capabilities for 2020: (1) Detecting and reporting threats or attacks against all space systems of national interest in near real time is critical; (2) The ability to withstand and defend against attacks on space systems; (3) Reconstituting and repairing space capabilities critical to national interests (if defenses fail) must occur within days, or even hours; (4) Assessing mission impact is critical to course of action development; (5) Identifying, locating, and classifying the source of a threat, ranging from intentional hostile acts to accidents or naturally occurring space events pose great challenges [9:33-34].*

The abilities to withstand and defend against attacks, preserve critical assets, and reconstitute and repair those critical assets are the primary focus of this research. Air Force Space Command is comprised of a network of critical nodes and links used to "provide real-time, survivable, and enduring communications, surveillance, environmental monitoring, navigation, and warning for unified and specified commanders, the national command authorities, and the intelligence community" [10:84]. This network of assets is among the military's "prime national security responsibilities" [10:70]. It must be protected. This requirement to protect comes from an objective to gain the control of space.

> *Control of Space requires USCINCSPACE to achieve five interrelated objectives: (1) assure the means to get to space and operate once there; (2) surveil the region of space to achieve and maintain situational understanding; (3) protect our critical space systems from hostile actions; (4) prevent unauthorized access to, and exploitation of, US and allied space systems and, when required, (5) negate hostile space systems that place US and allied interests at risk [9:20].*

Protecting the critical space systems is an objective Air Force Space Command must achieve. As the LRP states,

> *Our nation's increasing military and economic dependence on space power makes it likely for space to become a vital national interest. This same dependence also implies vulnerability. US interests and investments in space must be fully protected to ensure our nation's freedom of action in space [9:viii].*

Space Command must have "the capability to collect, process, and disseminate an uninterrupted flow of information while exploiting or denying an adversary's ability to do the same" [9:10]. In order to provide this capability the Air Force must be able to defend and protect the network of space assets. The Air Force must "develop advanced models and simulation capabilities to help analyze nodes, identify effects and determine which

capabilities we must rapidly reconstitute" [9:38]. These models will identify major components of the Space Command network of assets that are critical for operation.

**Statement of Problem**

The problem at hand is to develop a modeling and analysis capability to study nodes and links of a multi-disciplined network in Space Command. The analysis will determine which sets of nodes and links make the network most vulnerable to disruption. A multi-disciplined network consists of a network of components, which perform a primary function, and sub-network components, which perform secondary supporting functions. For example, the primary function of a communications network is communications, while the power station has the function of providing power to its corresponding component.

Analyzing the disruption of a multi-disciplined network begins with identification of a vulnerability set. This vulnerability set includes the links and nodes which could disrupt the network. The vulnerability sets are then analyzed to determine which are the most vulnerable and susceptible. Susceptible is referred to as vulnerabilities that are easily affected or disrupted. To assist in the process of vulnerability prioritization, a quantitative method is needed that will measure the significance or value of each vulnerability. Additionally, a method of finding and evaluating sub-network vulnerabilities is needed.

## Research Approach

This research uses a graph theory approach to portray the network and its major system components. A cut-set generation algorithm generates and identifies potential vulnerability sets comprised of the network's major components. Value-Focused Thinking (VFT) [16] develops a working value hierarchy to determine the objectives of the network. The hierarchy provides a structure, down to measurable attributes, which adequately define each objective. Assigned weights indicate each attributes relative importance within the hierarchy. Susceptibilities can be found using a value hierarchy which includes attributes that would define a susceptible component, for example, hardening or physical defense techniques.

Once the hierarchy is developed, the application creates a prioritized list of critical network vulnerabilities. Not only can this value hierarchy assess the components of the primary network but it can be used to assess vulnerabilities at the sub-network level, such as water and power. Thus, the analysis provides a comprehensive vulnerability assessment.

The cut-set algorithm and value model are accomplished via a *Network Vulnerability Assessment Tool* implemented in Visual Basic (VB). This package is designed to interface with an Excel spreadsheet in which the user inputs network and value hierarchy data. The tool was developed using Leinart's *Network Disruption Modeling Tool* [20] as a basis. New features of the software include an algorithm to find cut-sets of a directed network, sub-network analysis, and a routine to easily input new

value hierarchies. Additionally, the new software features sensitivity analysis to provide insight into the driving factors behind the vulnerability rankings. This analysis provides the decision maker an invaluable tool for finalizing the value hierarchy.

**Scope/Limitations**

This thesis focuses on a command and control network of Air Force Space Command components including sub-network components. The sub-network components consist of a variety of media types, including such items as water and power.

The components and characteristics of the system are the nodes and links of the network. Therefore they also represent candidate vulnerabilities. It is assumed network disruption is accomplished once command and control is lost between a designated source node(s) and a designated sink node(s).

For the purpose of this research, network disruption does not include corruption of actual data, but instead is concerned with the destruction or degradation of network components, which stop or hinder the network flow.

The network objectives and attributes, which are used in the hierarchy, will not be representative of all command and control networks and all decision makers' preferences. This research incorporates a hypothetical decision maker's preferences and demonstrates the methodology for a notional network.

## Thesis Overview

This chapter has presented a brief background of the problem to be addressed. Chapter II provides a review of current literature in the area of networks, graph theory, cut-set determination, and value-focused thinking. Chapter III gives the methodology used to obtain and quantify network cut-sets and objectives to identify possible network vulnerabilities. Chapter IV analyzes the results of a command and control network scenario. Chapter V contains conclusions from the research and recommendations for additional related research.

# II. Literature Review

## Introduction

The goal of this chapter is to provide the theoretical background necessary to model and analyze complex network vulnerabilities. This chapter provides the relevant terms, concepts, and exposure to attack this problem. Additionally, there is information about several network modeling tools.

## Networks

A network is "a system or pattern made up of interconnecting parts" [31:460]. There are many kinds of networks which all serve a different function. This research is interested in examining a network of multi-disciplined parts. In particular, it examines a system of parts, or assets, which make up a subsystem of Air Force Space Command. In this case the network may be represented as a system of links and nodes which work together to ensure space control.

Air Force Space Command has several networks. For example, the Air Force Satellite Control Network (AFSCN) is a network of nodes and links. The AFSCN is a global network system that provides command, control, and communications for space vehicles. The many nodes of the AFSCN consist of dedicated (supporting a single program) and common-user (supporting many programs) equipment and facilities which, collectively, provide operational telemetry, tracking, and commanding (TT&C) support for Department of Defense (DOD) space vehicles plus selected National Aeronautics and

Space Administration (NASA) and foreign allied nations' space programs. In addition, nodes are both mobile and fixed. Network elements include tracking stations, computer facilities, communication links, satellites, and control centers. All of these elements combined with many others make up the network of nodes and links that meet the requirement for constant control, support, and direction of satellites [10:73-74].

Another Space Command network is the space surveillance network (SSN). The SSN includes many radar and tracking devices which give the ability to monitor, assess, and inform of space activities, which is essential to space control. The SSN uses many worldwide sensors to perform its mission [10:97].

Another network, which is not so obvious, is the network of assets used to launch a space vehicle or provide space reconnaissance photos. In addition, many times the sub-networks providing power, water, sewer, and computer lines go unrecognized. This research considers how to assess a network comprised of many assets; not only communication, or computers, but also sub-networks providing water, power, communication, and many other nodes and links.

Network topology describes the layout and resources that model a physical network. Topology in this case consists of nodes to represent hardware and links to represent transmission media between nodes. Complex networks of nodes and links may be organized hierarchically into sub-nets [4].

The networks of interest will include sub-networks. A simple example of a sub-network is a network of water pumping stations and water lines. The pump stations are the nodes of the sub-network and the supply lines are the links.

Network vulnerability is defined in many ways. The focus here is disruption that occurs when elements of the network breakdown or are destroyed. A network is vulnerable if two nodes can be disconnected. If a few components can be taken out of the network and cause there to be no more operational paths between the two nodes, the network is vulnerable [2:682]. It is susceptible if the two nodes are "easy" to disconnect. Therefore, the underlying concept of vulnerability is connectivity.

**Graph Theory**

One way to model networks is through the use of graph theory. The network can be modeled by a graph. A graph consists of a finite set of points called nodes, connected by lines called links. Before proceeding, several definitions and concepts of graph theory need to be explained.

A graph that is in one piece is a connected graph; a graph in more than one piece is a disconnected graph. Both are shown in Figure 1. In addition, a graph is connected if it cannot be expressed as the union of two graphs, and disconnected otherwise [32:4,10].

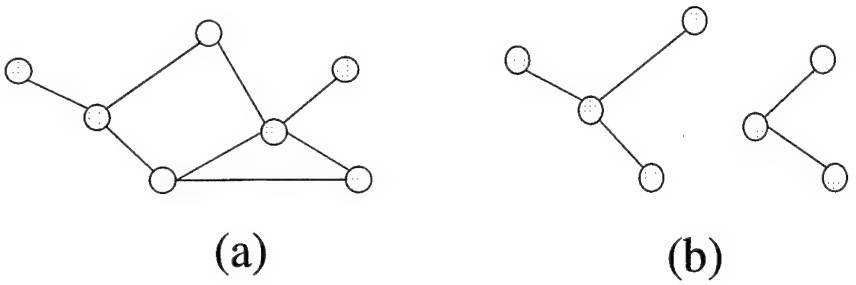


Figure 1 (a) connected simple graph (b) disconnected simple graph

A simple graph G consists of a set of elements called vertices (or nodes), and a set of distinct elements called edges. The set of vertices may be designated as V(G) and the set of edges may be designated as E(G) [32:8].

In a simple graph there is at most one edge joining a given pair of vertices. However, many results that hold for simple graphs can be extended to more general objects in which two vertices may have several edges joining them. Figure 2 shows a general graph. The term general graph or, simply, a graph, is used for graphs with multiple edges [32:8].

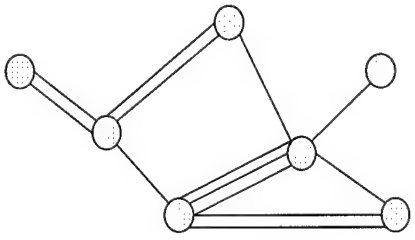

Figure 2 General Graph

Therefore, a general graph G consists of a set of elements called vertices, and a family of (not necessarily distinct) elements called edges. The use of the word 'family' allows multiple edges. The set of vertices V(G) and the *family* of edges E(G) [32:9].

Two vertices $v_1$ and $v_2$ of a graph G are adjacent if there is an edge $e_1$ joining them, and the vertices $v_1$ and $v_2$ are then incident with such an edge. Similarly, two distinct edges $e_1$ and $e_2$ are adjacent if they have a vertex in common [32:12]. Figure 3 shows an example of both adjacent vertices and edges.

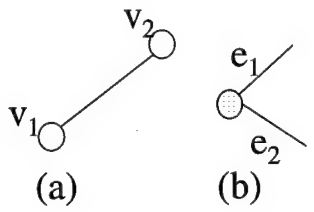


Figure 3 (a) adjacent vertices (b) adjacent edges

Cut-Sets

A disconnecting set in a connected graph G is a set of edges or vertices whose removal disconnects G. A cut-set is defined to be a disconnecting set, in which there is no proper subset. If G is a graph, a disconnecting set of G is a set of edges or vertices whose removal increases the number of components of G, and a cut-set of G is a disconnecting set, in which there is no proper subset [32:28-29]. In other words, a cut-set is a set of edges or vertices that if deleted, separate a connected graph into two or more components. The deletion of a proper cut-set results in precisely two components [3:29].

There are several kinds of cut-sets: vertex cut-set (also known as a separating set), where the vertices are removed (Figure 4(a)); edge cut-set, where the edges are removed (Figure 4(b)); and a mixed cut-set, where a combination of vertices and edges are removed (Figure 4(c)). The cut-set is the set of edges or vertices with the dotted line

drawn through them which separates the graph on the left into the disconnected graph on the right.

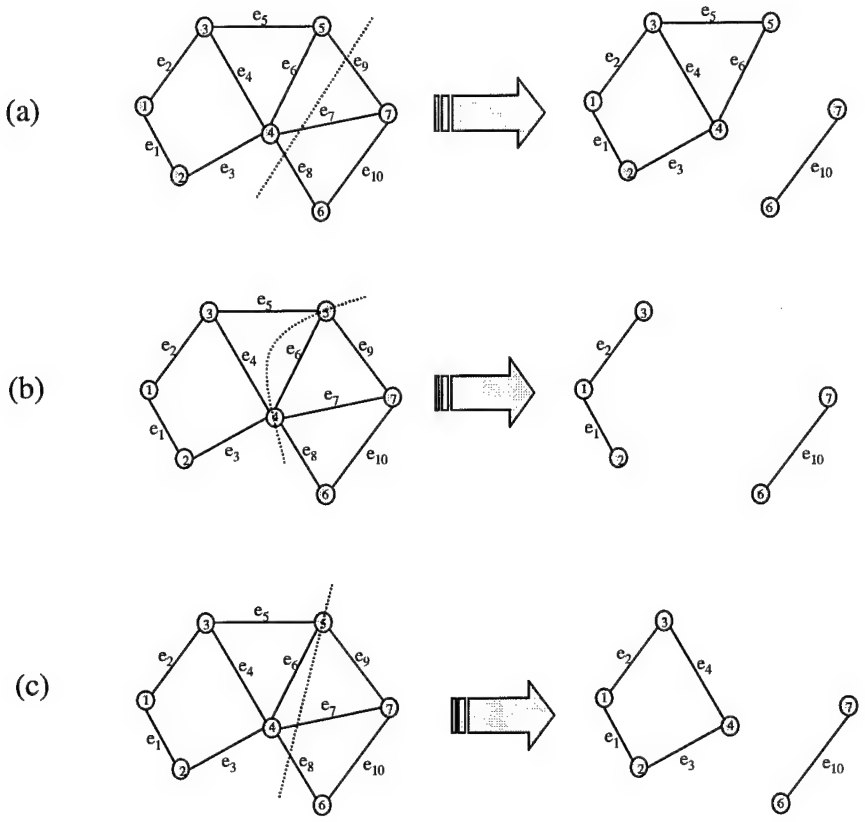


Figure 4 (a) edge cut-set (b) vertex cut-set (c) mixed cut-set.

If G is connected, its edge connectivity $\lambda(G)$ is the size of the smallest cut-set in G. Thus, $\lambda(G)$ is the minimum number of edges that need to be deleted to disconnect G [32:29].

A separating set, or vertex cut-set, is a set of vertices whose deletion disconnects G; note that when a vertex is deleted, the incident edges are also removed. If G is connected and not a complete graph, its (vertex) connectivity $\kappa(G)$ is the size of the

smallest separating set in G. Thus, $\kappa(G)$ is the minimum number of vertices that need to be deleted to disconnect G [32:29].

Directed Graphs

Graphs can be directed or undirected. Directed graphs occur when the edges have a direction from one vertex to another and only flow in one direction. Undirected graphs have no orientation and can flow in either direction.

Directed graphs, also called digraphs, are graphs that have the additional characteristic that every edge has been oriented or given a direction, as shown in Figure 5 [3:23,24]. A directed graph, D, consists of a set of elements called vertices, and a family of elements of called arcs. The set of vertices is V(D) and the family of arcs is A(D). A directed graph is connected if it cannot be expressed as the union of two directed graphs [32:100-101]. The arcs consist of a head, the termination of the arc, and a tail, the start of the arc. Thus, the heads of the arcs in Figure 5 are the points of the arrows.

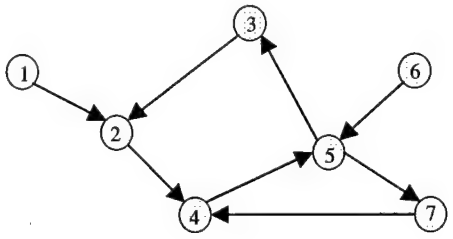


Figure 5 Directed Graph

In a directed graph the arcs of a cut-set can be classified into two groups: those directed from $v_1$ to $v_2$ and those directed from $v_2$ to $v_1$. Removal of the former set

disconnects all paths from $v_1$ to $v_2$, while removal of the latter disconnects all paths from $v_2$ to $v_1$ [3:30].

The study of graph theory and networks often involves the flow of information from one point to another point. In graph theory, the 'way of getting from one vertex to another' is defined as a walk. A path is a walk in which a vertex appears only once [32:3]. For example, a path in Figure 5 is 1→2→4→5→7. This is also defined as a path set of the network.

Also associated with directed graphs are the terms source and sink. Wilson defines a source as a vertex with 0 arcs coming into the vertex, or in-degree 0, and a sink as a vertex with 0 arcs going out, or out-degree 0 [32:105]. For the purpose of this thesis the source is simply the start of the data and the sink is the point in the network where the information will be terminated. In fact, there can be more than one source and/or sink.

Matrix Representation

Graph theory also uses matrices to represent network structure. Matrix representation is a convenient way to represent the structure of a graph. If G is a graph with n vertices and m edges, then the incident matrix is the $n \times m$ matrix whose rows and columns correspond to the vertices and edges. Whenever a vertex is incident with an edge, a 1 appears, as shown in Figure 6 [3:101-102].

$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

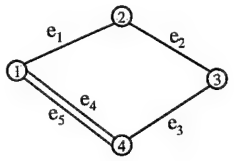


Figure 6 Incidence Matrix and Associated Graph

Another useful matrix representation of a graph is the cut-set matrix, each row of which characterizes one proper cut-set. In Figure 7, the proper cut-sets of a graph are shown.

$K_1 = \{e_2, e_3\}$ $K_5 = \{e_3, e_1\}$
$K_2 = \{e_1, e_2\}$ $K_6 = \{e_2, e_4\}$
$K_3 = \{e_1, e_4\}$ $K_7 = \{v_2, v_4\}$
$K_4 = \{e_4, e_3\}$ $K_8 = \{v_1, v_3\}$

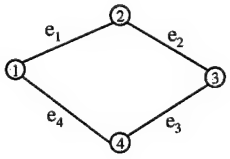


Figure 7 Cut-sets and Associated Graph

The cut-set matrix for Figure 7 is

$$K = \begin{array}{c} \\ K_1 \\ K_2 \\ K_3 \\ K_4 \\ K_5 \\ K_6 \\ K_7 \\ K_8 \end{array} \begin{array}{c} \begin{array}{cccccccc} v_1 & v_2 & v_3 & v_4 & e_1 & e_2 & e_3 & e_4 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

It is also useful to define an adjacency matrix, which represents the connected vertices. An adjacency matrix and its associated graph are shown in Figure 8.

$$V = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

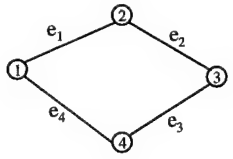


Figure 8 Adjacency Matrix and Associated Graph

Graph theory can be used to represent the structure of a network and to find vulnerabilities using cut-set algorithms.

**Value Focused Thinking**

After modeling and identifying all cut-sets of a network using graph theory, the cut-sets, or vulnerabilities, need to prioritized. Value Focus Thinking can be used to rank the vulnerability sets based on a decision maker's values.

Values

Value Focus Thinking begins by developing a value model to evaluate the alternatives. In doing so, tasks required to achieve the primary objective or goal and measures of merit that quantify the value of performing the task at different levels are identified. Top-level objectives describe aspirations that are most important to the decision maker. Objectives are decomposed until desired measures of merit can be specified. Weights are assigned to signify the relative importance of objectives at every level [8].

An important notion in VFT is value. Keeney says, "Values are what we care about. [Values] should be the driving force for our decision-making." Values are principles used for evaluation. Ethics, derived traits, characteristics of consequences that matter, guidelines for action, value trade-offs, and attitudes toward risk all indicate values [16:3-7]. The central role of thinking about values is illustrated in Figure 9.

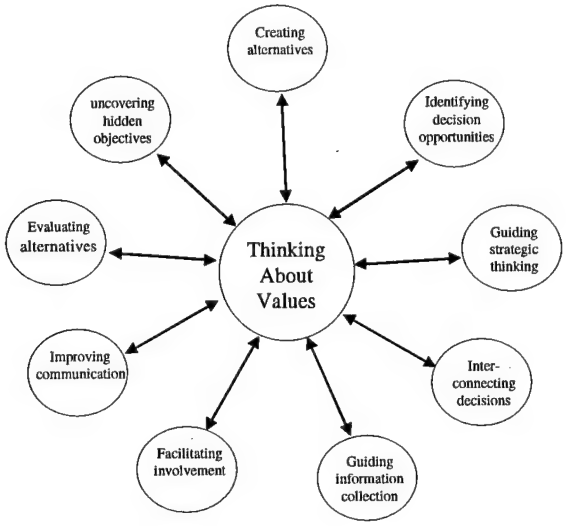


Figure 9 Overview of Value Focused Thinking [16:24]

In VFT, values are made explicit with objectives, and a hierarchical representation of objectives is constructed. The lower level objectives support the general objectives above them [8]. The objectives provide guidance for action and the foundation for analysis [16:33].

An objective is a statement of something that one wants to achieve. Three features characterize it: a decision context, an object, and a direction of preference. There are two types of objectives: fundamental objectives and means objectives. A fundamental objective states the essential reason for interest in the decision, a basis for

considering the decision. A means objective is the means to the achievement of the fundamental objectives; it implies the degree to which another objective can be achieved [16:34].

Value Model

A value model is the representation of the decision maker's objectives, tasks, subtasks, attributes, and scoring functions. The value model indicates the important information so alternatives can be evaluated more credibly. A value model is a branching structure, with the most fundamental objectives appearing at the top. Keeney states, "The higher-level objective is defined by the set of lower-level objectives directly under it in the hierarchy" [16:78]. Figure 10 shows the basic format for a value hierarchy.

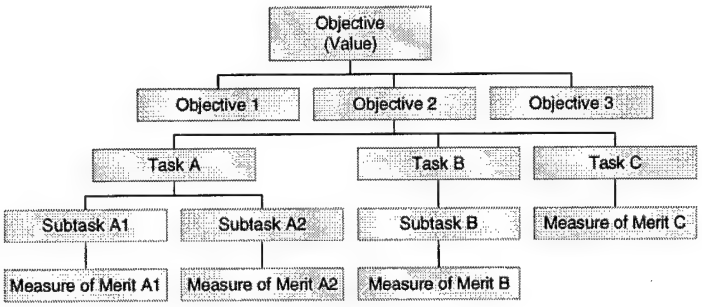


Figure 10 Value Model

A value model is a model with qualitative and quantitative relationships. The general procedure for building this model is essentially the same as for any model; principles, sound logic, reasoned judgments, and consistent data. The intent is to have the model lend some insight into a complex situation [16:130].

Clemens describes six specific characteristics of a value model: 1. complete, encompassing all important facets of the decision; 2. as small as possible; 3. objectives should only appear once; 4. objectives should be decomposable, and stand alone; 5. means and fundamental objectives should be distinguished; and 6. attribute scales must provide an easy way to measure performance or outcomes [6:533-534]. Combining the first, third, and fourth characteristics above yields two properties: the objectives must be mutually exclusive (only appear once and can be treated separately) and collectively exhaustive (encompass all that the decision maker values) [8]. Likewise, Kirkwood says that a value model should have the following characteristics: 1. evaluation considerations must adequately cover all concerns necessary to evaluate the overall objective; 2. non-redundancy: no two evaluation considerations should overlap; 3. evaluation considerations must be independent [17:16-18].

Attributes are metrics used to gauge system performance. Each attribute has a range, from worst to best. The metric is a measuring tool for determining to what degree an objective is achieved. An attribute should be measurable, operational, and understandable. Attributes that have those three properties will clarify the respective objectives and facilitate value-focused thinking. A critical issue affecting whether an attribute has those properties is ambiguity. Each attribute should be unambiguous, i.e. have a clear meaning. An attribute that is measurable explains the associated objective in more detail than the objective alone. It is operational if it describes the possible consequences of the associated objective and provides a basis for measuring the various degrees to which the objective might be achieved. To be understandable, there should be

no ambiguity in describing consequences and in interpreting consequences described in terms of attributes [16:100-116].

*Value Function*. The numerical rating that a decision alternative obtains with respect to a particular evaluation measure is called the score of an alternative [17:11-13]. The scoring function, provides a quantitative way to measure the performance of an alternative for each attribute. The value function is used to convert the scores into something meaningful to the decision maker. It is used to evaluate the performance of the alternative. The domain (horizontal axis) of the value function represents the attribute; the range (vertical axis) is the corresponding value score. A value function's domain may be quantitative or qualitative and may be different, but its range must be quantitative and the same. The development of these functions is an important task. Analysts must determine the decision makers' values for the full range of the attribute [8].

There are many types of value functions. One of the simplest is the linear value function. A linear function implies that each incremental increase in performance is valued just as much as the preceding increment [8]. It has the advantages of easy construction and computation. However, in many cases it is not warranted. For example, the value of an incremental increase in performance may be valued less than the preceding increment, known as diminishing marginal returns. Jackson says, "Another way of explaining diminishing returns is to say that some level of performance is 'good enough'" [8].

Another function used is increasing marginal returns, where an incremental increase in performance is valued more than the preceding increment. In this case

Jackson says, "the decision maker admits there is a certain threshold of performance that must be met before the system has substantive value" [8].

The S-curve function, which reflects a combination of the two previous examples of increasing returns then diminishing returns, is another value function used. The four examples given are the most common value functions; however, as Jackson states, "as long as the objective is to maximize value, any monotonically increasing representation is possible" [8].

Kirkwood provides a mathematical equation for an exponential single dimensional value function. This exponential function has a particular form that depends on the range of the measure and a constant called the exponential constant. This constant specifies the shape of the curve, thus allowing the marginal or diminishing returns. As the value of the constant increases, the graph will become less curved until finally the function becomes a straight line. When the constant is infinitely large, the function is exactly a straight line [17:65]. The following equations are Kirkwoods' exponential single dimensional value function, where $\rho$ is the exponential constant and the high and low terms specify the range of the measure being evaluated.

The equation (1) is for monotonically increasing functions where the higher amounts of x are preferred to lower amounts. The equation (2) is for monotonically decreasing where the lower amounts of x are preferred to higher amounts [17:65-66].

$$v(x) = \begin{cases} \dfrac{1-\exp[-(x-Low)/\rho]}{1-\exp[-(High-Low)/\rho]}, \rho \neq Infinity \\ \dfrac{x-Low}{High-Low}, otherwise \end{cases} \quad (1)$$

$$v(x) = \begin{cases} \dfrac{1-\exp[-(High-x)/\rho]}{1-\exp[-(High-Low)/\rho]}, \rho \neq Infinity \\ \dfrac{High-x}{High-Low}, otherwise \end{cases} \quad (2)$$

*Weights*. The next step is to assign weights to the different levels of the value hierarchy, thus showing the relative importance. There are several ways to assess the weights. Clemens suggests three different methods: pricing out, swing weighting and lottery weights. Pricing out is essentially determining the marginal rate of substitution between one attribute and any other, trying to find an indifference point. This technique becomes difficult to use when comparing two attributes with different value function types. Swing Weighting requires the decision maker to compare individual attributes directly by imagining hypothetical outcomes and swinging each attribute from its least preferred to its most preferred level. This technique can be used in almost any weight-assessment situation, but requires a lot of thought. The third type discussed is lottery weights. Lottery weights assess the probability that makes one indifferent between a lottery and the sure thing. The assessment is the weight for the one odd attribute in the sure thing [6:546-550]. The third type is not applicable since we aren't assessing probabilities.

The weights are assigned to the value hierarchy across each tier. They range between 0 and 1. For each task, the weights of the immediate subtasks must sum to 1.00. Figure 11 shows an example of weights assigned to a value model.

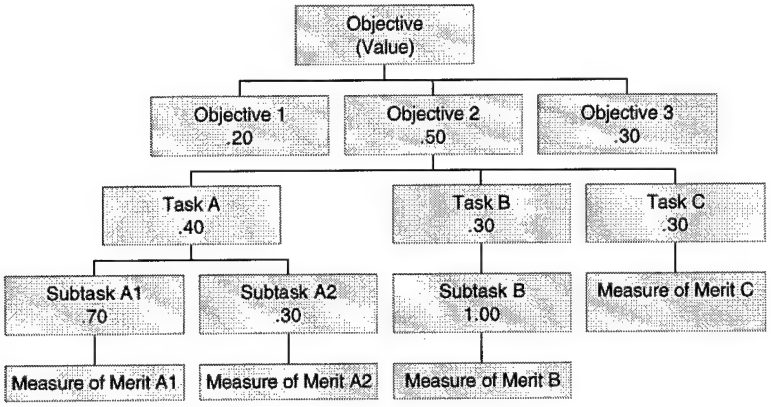


Figure 11 Value Model with Assigned Weights

*Additive Value Function*. If there are no interactions (mutual independence) among a set of attributes, a simple additive value function can be used to calculate a total score for an alternative. The additive utility function is $V(x_1,\ldots x_m) = k_1V_1(x_1) +\ldots+ k_mV_m(x_m)$ where the weights are $k_1,\ldots, k_m$, $x_i$ is the score of the attribute, and $V_i$ is the value function of the attribute [6:537].

After the value model is completed and weights have been assigned, the evaluation of alternatives can be accomplished. For each alternative, raw scores are evaluated using the applicable value functions. If a particular attribute does not apply to the alternative, the raw score is zero and hence the value of that attribute is 0. A total value is found by using the additive utility function. The alternative with the highest value is the "best" [8].

*Vulnerability Set Value Evaluation*. When an alternative is comprised of several components, as is the case of a vulnerability set, a way to calculate the overall value for the set is needed. There are several ways to approach calculating the vulnerability set value. One method of calculating the value is simply to take an average of the score for each component for each attribute and then apply the appropriate value functions. Another approach would be to take the individual component scores, apply the appropriate value functions, and then take the average of the values. Using the maximum or minimum score for the alternative would be another approach.

There are a large number of potential methods of evaluating the value of a vulnerability set. There are no known efforts that have explicitly studied which approach is the correct approach. For this research the method used averages the scores for each component and then applies the appropriate value function.

**Network Modeling Tools**

There are many algorithms and modeling tools for network analysis. The problem is finding one that satisfies the analysis requirements for the specific network at hand. Many of the network modeling tool's primary focus is communication or computer networks. For example, a computer program called Adversary is used at the National Air Intelligence Center (NAIC) for modeling communication networks. It is a great tool complete with graphical images; however, it is designed for a communication network. The program can be used to model different aspects of a communication network, but the

current version does not find cut-sets of the network. Therefore, it would be hard to find all the vulnerabilities of a network.

Another network analysis tool is called "NetWorker". This software was developed in Visual Basic and incorporates a graphical user interface (GUI) to display the network. The GUI is used to modify the network and then save it as a text file readable by the analysis algorithm [15]. However, one of the inputs was traffic demand. Additional information could not be obtained on whether or not this program was communication specific or if it could be used for any type of network. In addition, there was no information on the types of analysis the software is capable of doing.

Other network analysis tools include, but are not limited to: Tahoe Design Software for a pumps and fluid network [30]; Splice [12] and ESACAP [13] for an electrical circuit network; Novell's LANalyzer [25] and ODS Networks [26] for a computer network; NetSense [24] for internet analysis; Model Quest [21] and Brain Maker [5] for neuro networks; OPNET [23] for communication networks; and KrackPlot [19] for social networks. These are just a few of the many network modeling tools available today. However, there were no tools found that could model a generic multi-disciplined network and perform analysis to determine critical vulnerabilities of the network.

Cut-Set Algorithms

After researching various communication and computer network modeling tools, the focus changed to finding graph theory algorithms to generate cut-sets of a graph.

Like network modeling tools, there are a number of cut-set algorithms available for use. Hao developed "A faster algorithm for finding the minimum cut in a directed graph" [14:424]. This algorithm finds the minimum cut of a directed network without even specifying a source or sink node, and can easily be applied to undirected networks by replacing each undirected arc with two directed arcs. The algorithm finds the minimum cut for a network of nodes and arcs, where each arc of the network has an associated capacity.

Provan and Shier developed "A Paradigm for Listing (s,t)-Cuts in Graphs" [28:351]. An (s,t)-cut is a set which when removed disconnects a node s and a node t. An (s,t)-cutset is the minimal cut that does not contain any other cut. The paradigm can be used in a variety of graph types and has a time-per-cut complexity that is linear in the size of the graph.

*Network Disruption Modeling Tool*. Patvardhan, Prasad, and Pyara developed an algorithm for finding the vertex cut-sets of undirected graphs. The algorithm finds all s-t minimal vertex cut-sets separating two vertices, s and t [27]. This algorithm was used by Leinart to develop the Network Disruption Modeling Tool. Leinart's model not only generates cut-sets of a network but it also incorporates a tool for ranking the cut-sets using Value Focused Thinking [20].

Leinart uses the Patvardhan, Prasad, and Pyara algorithm to generate the cut-sets of a network. Leinart expanded the algorithm to find the vertex and link cut-sets for a network. He uses a transformation of a graph to represent the links as vertices and then applies the algorithm.

The cut-sets that are generated using the algorithm are then ranked using a value model developed using Value Focused Thinking techniques. The model is written in a Visual Basic/Excel Spreadsheet environment. Leinart's model, the Network Disruption Modeling Tool, is used to find and rank all the target sets of an enemy's communication network [20]. However, Leinart's model focuses on an undirected network, which would generate more cut-sets than needed in a directed network. In addition, only one source and one sink could be selected. In Space Command one network could include multiple sources and/or multiple sinks. The tool also requires code to be changed if a decision maker wants to change the values. Leinart's model is applied to a communication network with no sub-network interaction. The model does have the capability to rank the cut-sets using Value Focused Thinking. This model was the only model found which not only could find a cut-set but also already included a model to rank them [20].

*Network Vulnerability Assessment Tool*. The cut-set algorithm selected for this thesis effort was developed from MAXFLO, a program written in Fortran for Stochastic Network Performance analysis[1:3-1]. The program uses the concept of flow in a network. The information flows through capacitated arcs and nodes. In addition, the loss of an arc or node is equivalent to setting the capacity of the respective arc or node equal to 0 [1:3-4]. Thus a cut-set would consist of a path whose capacity is equal to 0.

Although the networks discussed in this thesis are not necessarily capacitated, this algorithm can be used to identify cut-sets by initially setting all node and arc capacities to 1. This allows each node and arc to be capacitated and set to 0 to disconnect them from the network.

In order to input the topology of the network into the *Network Vulnerability Assessment Tool* the network must be labeled using integers starting with one as the source. For the case of multiple source nodes, the first number of the network is reserved as a dummy node, where the actual source node follows the dummy node. Similarly the last number(s) are reserved for multiple sink node(s) followed by an additional number created for a dummy sink node. "Arcs are referred to by the source node integer called the Tail and the destination node integer called the Head" [1:3-11].

The algorithm by Shier and Whited used by Bailey provides a way to calculate cut-sets from pathsets by inversion [29]. As Bailey pointed out, "the best way to describe the algorithm is by an example network given in the article by Shier and Whited" [1:3-13]. The network is shown in Figure 12.

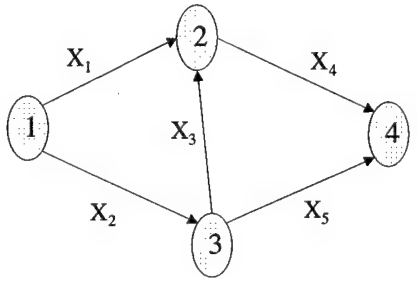


Figure 12 Directed Network with paths

A path polynomial is written as

$$X_1X_4+X_2X_3X_4+ X_2X_5$$

where $X_1$ $X_4$, $X_2$ $X_3$ $X_4$, and $X_2$ $X_5$ are path sets of the network. Therefore there are three paths from the source (s) to the sink (t). The inverse polynomial is obtained by complementing the polynomial giving

$$(X_1+X_4)(X_2+X_3+X_4)(X_2+X_5).$$

"If the inverse polynomial is expanded and the non-minimal elements are deleted, the result is the cut-set polynomial" [29]. Thus the next step is to expand the first two terms giving

$$(X_1X_2+X_1X_3+X_1X_4+X_1X_2+X_1X_3+X_1X_4+X_4X_2+X_4X_3+X_4X_4)(X_2+ X_5).$$

Next delete any elements that are contained in another element and eliminate duplicates giving

$$(X_1X_2+X_1X_3+X_1X_4+X_4X_2+X_4X_3+X_4)(X_2+X_5).$$

Finally expand the remaining elements and perform the same reduction techniques:

$$(X_1X_2X_2+X_1X_2X_5+X_1X_3X_2+X_1X_3X_5+X_1X_4X_2+X_1X_4X_5$$
$$+X_4X_2X_2+X_4X_2X_5+X_4X_3X_2+X_4X_3X_5+X_4X_2+X_4X_5).$$

Since $X_4X_2$ is contained in the $5^{th}$, $7^{th}$, $8^{th}$, and $9^{th}$ terms, $X_4X_5$ is contained in the $6^{th}$, $8^{th}$, and $10^{th}$ terms, and $X_1X_2$ is contained in the $2^{nd}$ and $3^{rd}$ terms, the equation is reduced to

$$(X_1X_2+X_1X_3X_5+X_4X_2+X_4X_5),$$

which is the cut-set polynomial. Thus the network has four cut-sets: $X_1X_2$, $X_1X_3X_5$, $X_4X_2$, and $X_4X_5$.

The cut-sets that are generated using the Shier and Whited algorithm are then ranked using a value model developed using Value Focused Thinking techniques, similar

to what Leinart [20] used. This new technique has the ability to find vulnerability sets in a directed network with any number of sources and sinks. In addition, it uses a value model to rank the vulnerabilities.

**Summary**

The theory presented in this chapter establishes the concepts used for this research effort. The key points that should be noted are: 1) the vulnerability of a network is the disruption of elements in the network; 2) graph theory can be used to represent and analyze a network, finding cut-sets of the network; 3) value focused thinking can then be used to rank the cut-sets by using a value model to represent a decision makers values; and 4) many network modeling tools are available, however, the key is to find the one that best models and analyzes the network of interest. The next chapter uses these concepts to develop a methodology for analyzing the multi-disciplined network of interest.

# III. Methodology

## Background

This research uses a graph theory approach to portray a multi-discipline network. The network components are represented as nodes and links of a graph. A Network Vulnerability Assessment Tool is used to analyze the network's vulnerabilities. The first step identifies what makes a link or node critical and then assesses the links and nodes. A value model is developed to assess the criticality of links and nodes. Specifically, the value model determines the relative importance of the vulnerabilities. The model is implemented in the Network Vulnerability Assessment Tool. Figure 13 shows the steps of the methodology, the models that are used at each step, and the outputs of each step. The methodology starts with the configuration of the network to be analyzed and ends with a prioritized list of network and sub-network vulnerabilities. Each step is accomplished in an Excel spreadsheet environment. The Network Vulnerability Assessment Tool was developed using Leinart's Network Disruption Modeling Tool [20] as a basis. A new cut-set algorithm has been incorporated to give the capability of analyzing a directed network with the possibility of multiple sources and/or sinks. Other modifications to the program have added the capabilities discussed throughout the methodology, including the sub-network routine.

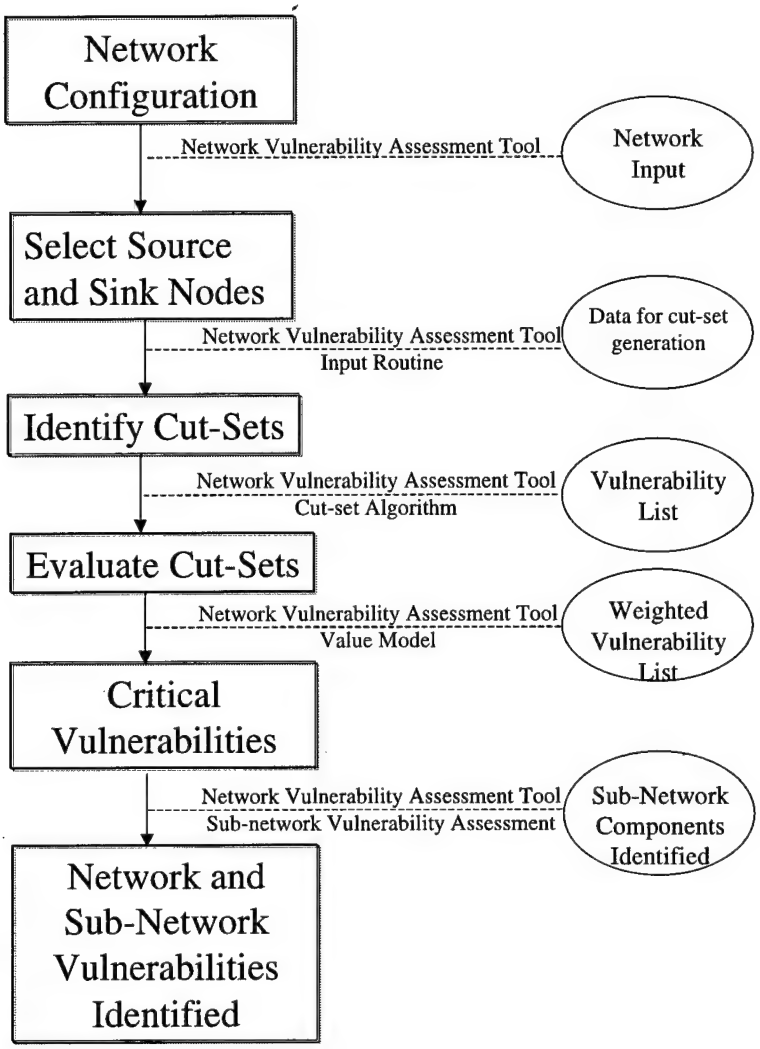


Figure 13 Flow Chart of Methodology

**Network Configuration**

The methodology begins by defining the configuration of the network to be analyzed. This configuration describes each component of the network, including all links and nodes. The topology of the network is inserted into the model by the user. The user specifies how many nodes are in the network, whether the source or sink is a dummy node which allows the use of multiple sources and sinks, and the links by specifying the

tail and head of each. In addition, the characteristics of each component are input. The characteristics are used later in the value model.

To illustrate a network configuration, an example is presented. Figure 14 is a graphical representation of the network topology of a generic network.

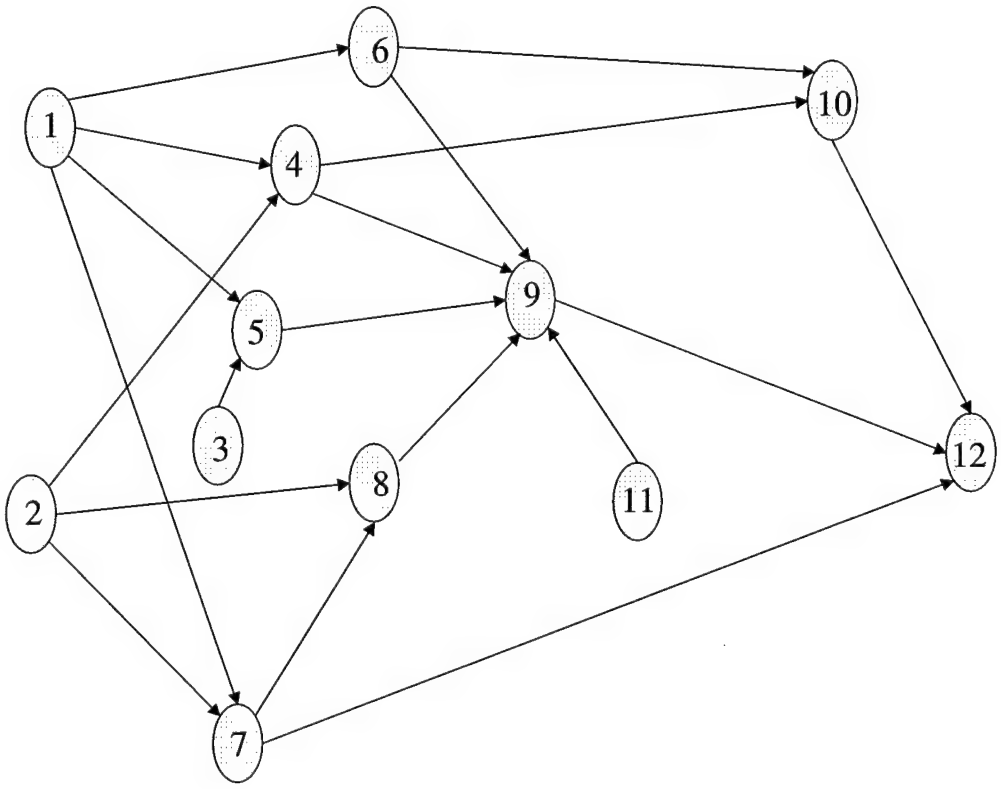


Figure 14 Generic Network Topology

The topology can be input into the Network Vulnerability Assessment Tool for further analysis. The total number of nodes is specified as 12. The source and sink nodes 1 and 12 respectively are not dummy nodes, and are entered as such. The links are then individually entered into the model. For example, the link from 1 to 7 is input into the model as 1 being the tail of the arc and 7 being the head of the arc. Similarly, the rest of the links are entered.

Now that the network topology has been represented, the specific characteristics of each link and node must be defined. As an example, the example network shown is defined as the network of nodes and links used to make breakfast, specifically oatmeal and toast. The nodes will represent the stove, refrigerator, sink, cooked meal, store, etc. The links represent the movement of something, for example, the movement of the butter from the store to the refrigerator. The entire description of the sample network and the characteristics of each node and link are shown in Appendix B.

The characteristics of the network are input into an Excel spreadsheet using the Visual Basic interface. The Excel environment is used to establish a source to store and organize the data of the network.

Finding a cut-set in the making breakfast scenario would provide little value to the analysis due to the nature of links. Each link could potentially be a vulnerability that would disrupt the network. Combining links or nodes into a cut-set would only give the impression that several links would have to be eliminated instead of just one to disrupt the network. Due to the insignificance of finding a cut-set in the making breakfast scenario, the generic network will be the focus in the next step of the methodology, cut-set generation.

**Cut-set Generation**

This step involves identifying the vulnerability sets. A cut-set algorithm written in visual basic and implemented in the Network Vulnerability Assessment Tool performs the cut-set generation. The cut-sets are defined as vulnerability sets.

The Network Vulnerability Assessment Tool uses the inversion method, discussed in the previous chapter, to generate the cut-sets. The algorithm uses the network topology and determines all the path-sets of the network. It then inverts the path-sets which generates the cut-sets of the network. These cut-sets are identified as the vulnerability sets.

As an example, the generic network in Figure 14 is defined as a directed graph, G(12,21) with 12 vertices and 21 edges. If vertex 1 and 12 are designated as the source and sink (the two nodes between which disruption is desired) then the Network Disruption Modeling Tool generates 352 vulnerability sets, and seven paths.

Table 1 Vulnerability Sets for Generic Network

| | | | | |
|---|---|---|---|---|
| Vulnerability Set 1 | Link 1 to 4 | Link 1 to 6 | Link 1 to 7 | Node 9 |
| Vulnerability Set 2 | Link 1 to 6 | Link 1 to 7 | Node 4 | Node 9 |
| Vulnerability Set 3 | Link 1 to 4 | Link 1 to 6 | Link 1 to 7 | Link 9 to 12 |
| Vulnerability Set 4 | Link 1 to 6 | Link 1 to 7 | Node 4 | Link 9 to 12 |
| Vulnerability Set 5 | Link 1 to 6 | Link 1 to 7 | Link 4 to 10 | Node 9 |
| Vulnerability Set 6 | Link 1 to 6 | Link 1 to 7 | Link 4 to 10 | Link 9 to 12 |
| Vulnerability Set 7 | Link 1 to 4 | Link 1 to 7 | Node 6 | Node 9 |
| Vulnerability Set 8 | Link 1 to 7 | Node 4 | Node 6 | Node 9 |
| Vulnerability Set 9 | Link 1 to 4 | Link 1 to 7 | Node 6 | Link 9 to 12 |
| Vulnerability Set 10 | Link 1 to 7 | Node 4 | Node 6 | Link 9 to 12 |
| Vulnerability Set 11 | Link 1 to 7 | Link 4 to 10 | Node 6 | Node 9 |
| Vulnerability Set 12 | Link 1 to 7 | Link 4 to 10 | Node 6 | Link 9 to 12 |
| Vulnerability Set 13 | Link 1 to 4 | Link 1 to 7 | Link 6 to 10 | Node 9 |
| Vulnerability Set 14 | Link 1 to 7 | Node 4 | Link 6 to 10 | Node 9 |
| Vulnerability Set 15 | Link 1 to 4 | Link 1 to 7 | Link 6 to 10 | Link 9 to 12 |
| Vulnerability Set 16 | Link 1 to 7 | Node 4 | Link 6 to 10 | Link 9 to 12 |
| Vulnerability Set 17 | Link 1 to 7 | Link 4 to 10 | Link 6 to 10 | Node 9 |
| Vulnerability Set 18 | Link 1 to 7 | Link 4 to 10 | Link 6 to 10 | Link 9 to 12 |
| Vulnerability Set 19 | Link 1 to 7 | Node 9 | Node 10 | |
| Vulnerability Set 20 | Link 1 to 7 | Link 9 to 12 | Node 10 | |

The vulnerability sets are a complete collection of the minimal set of vertices and/or edges, whose removal from the network will divide the network into two sub-graphs, with vertex 1 in one component and vertex 12 in the other. Twenty of the vulnerability

sets are shown in Table 1. The vulnerability sets are used in the value model to prioritize and generate the most vulnerable sets of the network.

**Vulnerability Set Evaluation**

Value-Focused Thinking [16] is used to prioritize the vulnerability sets, which were generated using the cur-set algorithm. First a value model is selected or defined using existing Air Force documents and/or a decision maker's values. The value model is a working value hierarchy to determine the objective(s). The hierarchy is structured down to measurable attributes that adequately define each objective. Assigned weights indicate each attribute's relative importance within the hierarchy. The vulnerability sets are then ranked using the value model. The final result generates a prioritized vulnerability list.

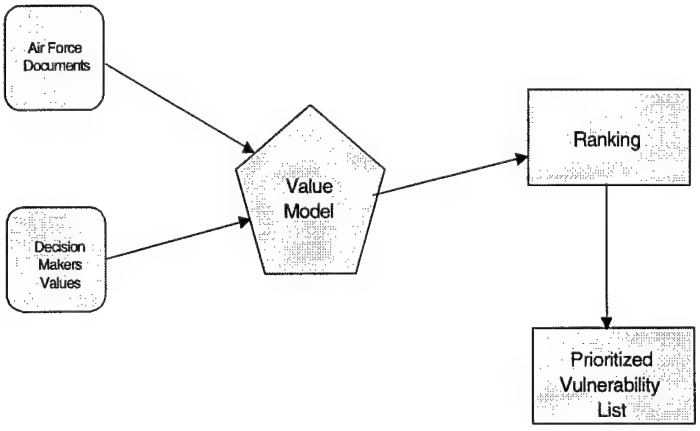


Figure 15 Value Process

Value Model Development

The value model is used to rank or score the vulnerability sets. In order to correctly and efficiently rank a vulnerability set several characteristics of the nodes and links need to be known. But even before characteristics are assessed, the fundamental objective(s) must be stated.

As mentioned in the introduction chapter, Air Force Doctrine states that the objective is to gain and maintain space control. As was further discussed, USCINSPACE has several objectives to achieve space control. One of those objectives is to "protect our critical space systems from hostile actions" [9:20]. This seems to be the focus throughout several space documents such as Air Force Doctrine Document 2-2 [11], Space Command's Long Range Plan [9], and the Space Handbook: A War Fighter's Guide to Space [10]. In addition, counterspace is mentioned as a means of gaining and maintaining this control. Counterspace is then broken down into offensive and defensive actions. Figure 16 shows several characteristics of those actions used to control space.

The defensive aspect is the primary focus. Defensive counterspace may either be active or passive. Passive counterspace defense is the area of interest in this research effort. Passive defense is broken down into two categories in Air Force Doctrine Document 2-2: reduce vulnerabilities, and, protect and increase survivability. The next step in finding the fundamental objective is to reduce this even further, focusing in on reducing vulnerabilities.

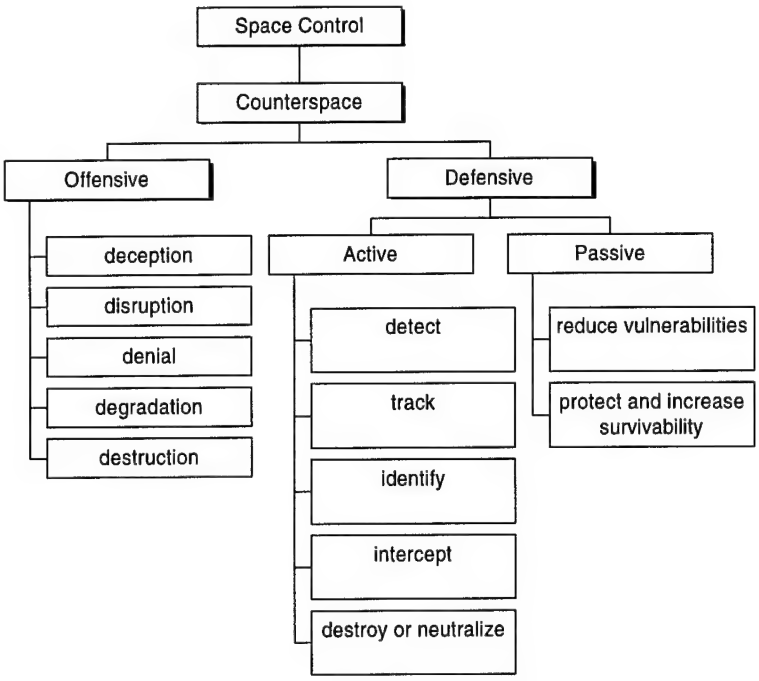


Figure 16 Space Control Characteristics

There are several steps to gaining space control by reducing vulnerabilities. Figure 17 shows a logical flow of reducing vulnerabilities. In order to reduce the vulnerability, the vulnerability must be known, assessed, and fixed. Passive defense starts with understanding allied networks and finding the weak links. Modifying the weak links allow the survivability of the network to be increased. Using a value model, the vulnerabilities can be ranked to find the most critical to the network.

Another viewpoint is also important to consider. For example, what is the objective of an adversary? The adversary wants to deny, disrupt, or destroy the network. The goal is to find the vulnerabilities of the network that could pose as the adversary's potential targets. This will help in protecting assets from hostile threats.

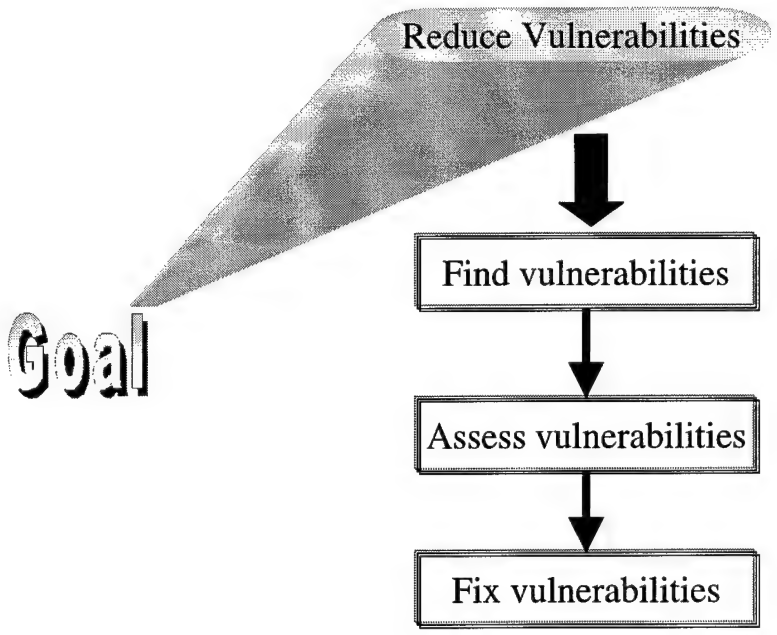


Figure 17 Vulnerability Analysis

Before finding and assessing the vulnerability of a network structure, the network requirements need to be clearly stated and understood. Several sources provide guidance in this area and are shown below. "The battle space information system…will provide: A redundant, seamless network of cross service and interagency links; Secure, and responsive…; Accurate and timely…" [9:10]. The space network should be an uninterrupted flow of information [9:10]. The network of space assets must have the capability to be reconstituted and repaired within days, or even hours [9:3]. The space systems include capabilities that are real-time, survivable, and enduring [10:84]. "Due to emerging threats to space assets; it is critical to maintain redundancy among terrestrial and space systems" [11:19]. In addition, a network should be robust, flexible, and sustainable [9:21-24].

Using the requirements of the system, a value hierarchy is constructed. The vulnerability value consists of items that could cause a system to be vulnerable or indestructible. It incorporates the key characteristics of the network system. In addition, it includes characteristics which would cause the network to be susceptible to disruption. For example, a network with no physical defense would be susceptible, or 'easy' to disrupt. Figure 18 shows a possible value hierarchy, with operability, survivability, and flexibility as key objectives. It is then further broken down into the characteristics of a space network as discussed earlier, with a few additions.

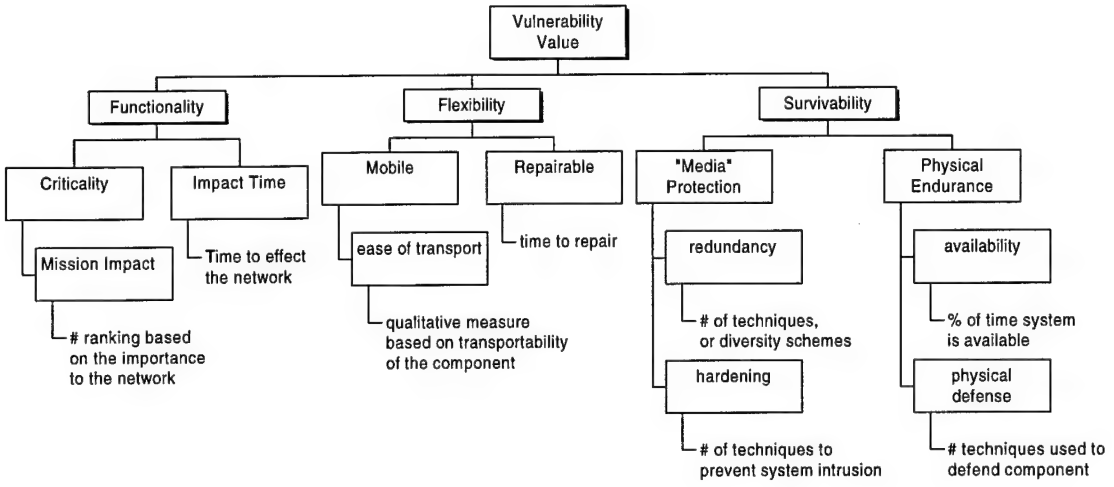


Figure 18 Value Hierarchy

Functionality determines the effects of a stressed environment to the network. Flexibility is the ability to adapt to a stressed environment under duress. Survivability is the ability to perform in a stressed environment during duress. These objectives or goals are used to score the vulnerability sets. The sets are ranked according to its criticality to the operation of the network. Network operation provides a key feature for space control. A detailed explanation of the value hierarchy is found in Appendix A.

Vulnerability Set Value Model Evaluation

Once the objectives and goals have been defined and a value hierarchy has been constructed, the Network Vulnerability Assessment Tool is used to evaluate the cut-sets. The vulnerability set's importance is obtained by scoring the set against each measure using an average of the score for each component in the set, then evaluating each score using the appropriate value function, and finally summing the weighted values. Each measure has an associated weight. The value of each measure is multiplied by the weight and "rolled-up" into the next layer of the hierarchy until the overall objective is reached. This produces the overall vulnerability score for the set of links and nodes. Equation 3 is the vulnerability set value function, where $w_i$ is the weight of the evaluation measure, $x_{ij}$ is the evaluation measure score for each component in the vulnerability set, $s_{ij}$ is the evaluation measure scoring function for each component in the vulnerability set, $V_i$ is the value function of the evaluation measure, and n is the number of components in the vulnerability set.

$$Value = \sum_i w_i V_i \left( \frac{\sum_j^n s_{ij}(x_{ij})}{n} \right) \qquad (3)$$

The score of each measure is based on the characteristics of the network input into the model and the measure's value function. The value function is used to normalize the measures so an additive value function may be applied. A decision maker determines the value function for the network of interest and inputs the function into the Network Vulnerability Assessment Tool.

As an example of the value model, we use the breakfast preparation example. Each node and link is defined as a vulnerability set. In other words, if the stove is removed the network is disrupted. Similarly, if the movement of butter from the store to the refrigerator is removed the network is disrupted. The importance of each of the nodes and links, or vulnerability sets, to the network is of interest. Using the characteristics and weights of the network in Appendix B, the measures described in Appendix A, and simple linear value functions, scores for each node and link are assessed using Equation 3. The scores are ranked from highest to lowest, thus producing a prioritized list of the network vulnerabilities. The scores and prioritized list are shown in Appendix B. The decision maker can truncate the list of vulnerabilities, for greater manageability. The Network Vulnerability Assessment Tool allows the user (or decision maker) to define the size of the list.

**Sub-Network Vulnerabilities**

A vulnerability list for the network is constructed based on the theory of cut-sets. However, many times in a network there are nodes or links that are not part of a cut-set that can disrupt the network. For example, a node that is part of a vulnerability set can be disrupted by loss of power. The loss of power can be caused by the disruption of a power supply unit. This power supply unit may not appear in the cut-sets generated. For example, a power supply unit in the local city could cause preparation of breakfast to be disrupted. Thus a technique to find vulnerabilities in the sub-networks must be generated.

There are several methods to approach this problem. The methodology presented for finding cut-sets using graph theory could be ignored and the entire network of links and nodes could be examined and ranked by the value model alone. The problem with this technique is quickly recognized in large networks. Every component is considered a vulnerability and must be examined. The next step is to figure out which of the components would actually disrupt the network. This could be a very tedious process.

Another approach is to find another cut-set algorithm that does not require the use of a source or sink node. It simply finds all possible cut-sets of the network, any vertex or edge that would separate the network into two components. This process would generate most of the vulnerabilities. However, if a node is attached to only one edge, the node itself will not be considered a cut-set. The definition of a cut-set does not allow for a single node to be considered a cut-set. This approach not only requires finding an algorithm that could generate such cut-sets, but also requires each node that is attached to only one edge be added or included in the analysis separately.

Using the Network Vulnerability Assessment Tool, a list of vulnerability sets could be generated using each node as a source and sink node, deleting any duplicates. The tool would be run in a loop until every node was a source and a sink. Using every node as the source and sink node would generate all possible cut-sets. This approach would include the links and nodes of a sub-network; however, the processing would be time consuming.

Based on availability, processing time, and compatibility with the Network Vulnerability Assessment Tool, the selected approach is now described. The decision

maker determines the manageable size of the vulnerability list produced by the modeling tool. Using this list, the nodes are extracted and any sub-networks are defined. Each node and link of the sub-networks becomes a possible vulnerability. Each is scored and ranked using the same value model techniques used in ranking the vulnerability sets. The Network Vulnerability Assessment Tool ranks all the sub-network components together. However, vulnerabilities of individual nodes can easily be extracted, giving a prioritized list for each critical node. For example, in Figure 14 node 3 and link 3 to 5 are considered sub-networks of node 5. During the scoring and ranking process, node 3 and link 3 to 5 will be ranked against one another. Node 12 and link 9 to 12 will be ranked against one another and assigned as a sub-network to node 9. These nodes and links will not be generated as a part of a cut-set unless the source or sink node is node 3 or node 12.

The sub-network vulnerability generation allows a decision maker to distinguish which components of sub-networks can effect the entire network. The components are listed according to which node they will effect in the network.

**Summary**

The most vulnerable set of nodes and/or links of a network are generated using the Network Vulnerability Assessment Tool. This tool incorporates an algorithm that produces the cut-sets of the network of interest. The cut-sets are transformed to vulnerability sets and evaluated using a defined value model. The decision maker creates this value model using value focused thinking techniques. The total values are ranked from highest to lowest producing a prioritized list of vulnerability sets. The top

vulnerability sets are used to find sub-network vulnerabilities. The sub-network vulnerabilities are evaluated using a value model (possibly the same as the network itself) and value focus thinking techniques. Finally, the methodology is implemented in a Network Vulnerability Assessment Tool written in an Excel/Visual Basic environment. The methodology provides a systematic way of determining the most vulnerable components of a network.

# IV. Results and Analysis

This chapter presents analysis for a notional network using the methodology developed in the previous chapter. The analysis includes results obtained from the *Network Vulnerability Assessment Tool.*

## Network Vulnerability Assessment Tool

The *Network Vulnerability Assessment Tool* is a self-driven tool written in a Visual Basic/Excel environment. The tool is used to generate, score and rank the vulnerability sets and then perform sensitivity analysis, persistency analysis, and occurrence analysis. In addition, the tool provides a separate sheets for evalauting the sub-network components. The Visual Basic code for each step can be found in Appendix D Visual Basic Code.

## The Notional Network

The network presented in this chapter is a hypothetical representation of Air Force Space Commands (AFSPC) Ground-based Electro-Optical Deep Space Surveillance (GEODSS) sensor network. The notional network consists of eighteen nodes and twenty-seven links as shown in Figure 19. Node 1 and Node 20 represent a dummy source and a dummy sink respectively. Adding these nodes allows analysis of multiple sources and sinks. An assessment of the vulnerabilities, which would cause a disruption of data flow from nodes 2, 3, and 4 to nodes 18 and 19, is desired. Disrupting the data flow between these nodes would hypothetically cut off all deep space optical sensor data to AFSC and

Navy Space Command (NAVSPACECOM). Network components are described in Table 2 and Table 3. Network attribute data are listed in Appendix C Notional Network Data.

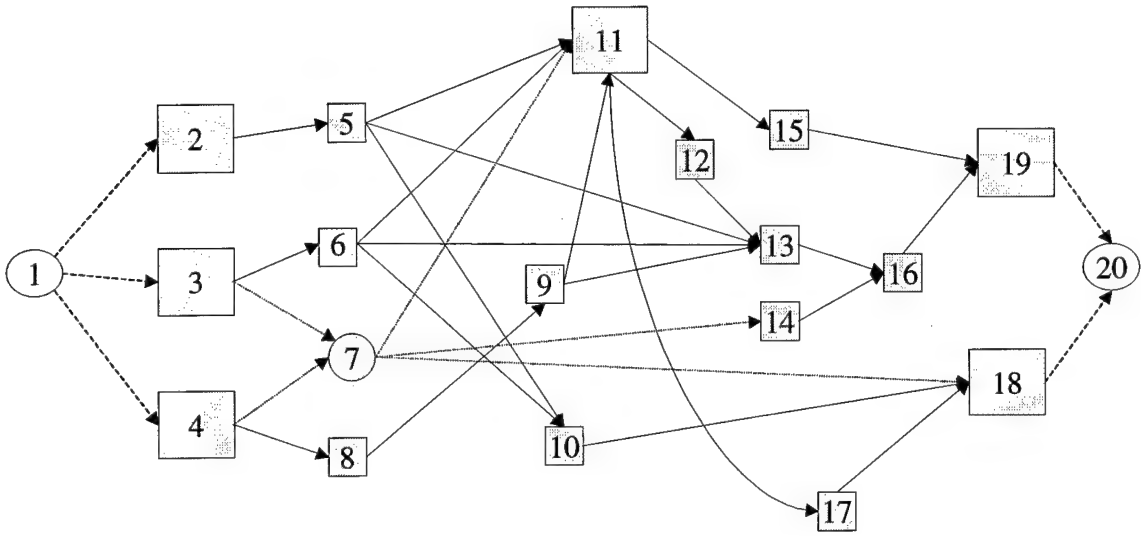


Figure 19 Notional Network

Table 2 GEODSS Network Nodes

| Components | Description |
|---|---|
| Node 2 | Socorro, New Mexico Optical Sensor |
| Node 3 | Maui, Hawaii Optical Sensor |
| Node 4 | Diego Garcia, British Indian Ocean Territories Optical Sensor |
| Node 5 | Primary Router |
| Node 6 | Primary Router |
| Node 7 | Satellite |
| Node 8 | Primary Router |
| Node 9 | Primary Router |
| Node 10 | Tertiary Router |
| Node 11 | Optical Command Facility (OC3F) – Edwards Air Force Base |
| Node 12 | Primary Router |
| Node 13 | Peterson Air Force Base |
| Node 14 | Shriever Air Force Base |
| Node 15 | Secondary Router |
| Node 16 | Colorado Springs Primary Router |
| Node 17 | Tertiary Router |
| Node 18 | Navy Space Command (NAVSPACE) |
| Node 19 | Space Control Center - Cheyenne Mountain Air Station (CMAS) |

Table 3 GEODSS Network Links

| Component | Description |
|---|---|
| Link 2 to 5 | Primary Data Line |
| Link 3 to 6 | Primary Microwave Line |
| Link 3 to 7 | Satellite Relay |
| Link 4 to 7 | Satellite Relay |
| Link 4 to 8 | Primary Microwave Line |
| Link 5 to 10 | Tertiary Data Line |
| Link 5 to 11 | Primary Data Line |
| Link 5 to 13 | Secondary Data Line |
| Link 6 to 10 | Tertiary Data Line |
| Link 6 to 11 | Primary Data Line |
| Link 6 to 13 | Secondary Data Line |
| Link 7 to 11 | Satellite Relay |
| Link 7 to 14 | Satellite Relay |
| Link 7 to 18 | Satellite Relay |
| Link 8 to 9 | Primary Data Line |
| Link 9 to 11 | Primary Data Line |
| Link 9 to 13 | Secondary Data Line |
| Link 10 to 18 | Tertiary Data Line |
| Link 11 to 12 | Primary Data Line |
| Link 11 to 15 | Secondary Data Line |
| Link 11 to 17 | Tertiary Data Line |
| Link 12 to 13 | Primary Data Line |
| Link 13 to 16 | Primary Data Line |
| Link 14 to 16 | Secondary Data Line |
| Link 15 to 19 | Secondary Data Line |
| Link 16 to 19 | Primary Data Line |
| Link 17 to 18 | Tertiary Data Line |

In addition to the data flow network, the sub-network analysis is performed for nodes 2, 3, 4, 11, 18, and 19. This sub-network analysis includes power and water stations and respective supply lines to support the listed nodes. The sub-network components are described in Table 4. The sub-network attribute data are listed in Appendix C Notional Network Data.

Table 4 Sub-Network Components

| Component | Description |
|---|---|
| ***Node 2*** | Socorro Optical Sensor |
| 2P1 | Socorro Power Station |
| 2W1 | Socorro Water Station |
| 2W2 | Socorro Water Station |
| Link 2P1 to 2 | Socorro Power Line |
| Link 2W1 to 2W2 | Socorro Water Line |
| Link 2W2 to 2 | Socorro Water Line |
| ***Node 3*** | Maui Optical Sensor |
| 3P1 | Maui Power Station |
| 3P2 | Maui Power Station |
| Link 3P1 to 3P2 | Maui Power Line |
| Link 3P2 to 3 | Maui Power Line |
| ***Node 4*** | Diego Garcia Optical Sensor |
| 4P1 | Diego Garcia Power Station |
| 4W1 | Diego Garcia Water Station |
| Link 4P1 to 4 | Diego Garcia Power Line |
| Link 4W1 to 4 | Diego Garcia Water Line |
| ***Node 11*** | OC3F – Edwards AFB |
| 11P1 | Edwards AFB Power Station |
| Link 11P1 to 11 | Edwards AFB Power Line |
| ***Node 18*** | NAVSPACE |
| 18P1 | NAVSPACE Power Station |
| 18W1 | NAVSPACE Water Station |
| Link 18P1 to 18 | NAVSPACE Power Line |
| Link 18W1 to 18 | NAVSPACE Water Line |
| ***Node 19*** | Cheyenne Mountain Air Station |
| 19P1 | CMAS Power Generator |
| 19P2 | Colorado Springs Power Station |
| 19P3 | Colorado Springs Power Station |
| 19W1 | Colorado Springs Water Station |
| 19W2 | CMAS Water Station |
| Link 19P1 to 19 | CMAS Power Line |
| Link 19P2 to 19P3 | Colorado Springs Power Line |
| Link 19P3 to 19 | CMAS Power Line |
| Link 19W1 to 19W2 | Colorado Springs Water Line |
| Link 19W2 to 19 | CMAS Water Line |

The primary objective of the GEODSS sensor network is to collect deep space object identification and metric position data. Nodes 2, 3, and 4 collect the data and transfer it to the Optical Command Facility (OC3F) at Edwards Air Force Base. The OC3F is represented as Node 11. This data is then sent to Peterson Air Force Base where

the space object identification (SOI) data is extracted. Peterson Air Force Base is represented as Node 13. Finally, the metric position data is sent to the Space Control Center (SCC) at Cheyenne Mountain Air Station (CMAS), represented as Node 19.

Secondary routes are available for the flow of data through the network. These secondary routes are made available by other routers and links in the network. In addition, NAVSPACECOM acts as the alternate SCC (ASCC) in case CMAS is unavailable. Therefore, to totally disrupt the network all paths must be cut off between nodes 2, 3 and 4, designated as the sources, and nodes 18 and 19, designated as the sinks. Due to both the primary and secondary routes and the alternate SSC, there are 24 paths from the sources to the sinks. Each one of these paths must be cut to disrupt the network flow. This disruption would in turn reduce the reliability of the network to zero. Reliability in this case is defined as one if there is at least one path from the source nodes to the sink nodes. If there is no longer a path between the source and sink, the networks reliability becomes zero. This causes the network to be inoperable.

Using the cut-set algorithm in the *Network Vulnerability Assessment Tool,* a list of possible ways to disrupt the network are found. For the notional network, the cut-set algorithm yields 34,285 cut-sets. The cut-sets represent the potential ways of reducing the reliability of the network to zero. In other words, there are 34,285 different ways of cutting off every path from the sources to the sinks. Thus, these cut-sets become potential vulnerabilities of the network. Appendix C Notional Network Data contains the top twenty vulnerability sets. The code used to generate the vulnerability sets is given in Appendix D Visual Basic Code.

The next step is to prioritize the vulnerability sets. The value model is used to evaluate and prioritize the 34,285 vulnerability sets.

**Value Model Results**

The value model consists of a hypothetical multi-attribute value function structured from the decision maker's value hierarchy. This value function was developed using the methodology discussed in the previous chapter, beginning with the overall objective of space control. The value hierarchy was approved and reviewed by AFSPACECOM/DOIK. The final value hierarchy, along with its weights, is shown in Figure 20. The three major objectives defining network vulnerability are functionality, flexibility, and survivability. These three objectives are further divided into sub-objectives and finally attributes or evaluation measures. The hierarchy contains eight evaluation measures: criticality, impact time, ease of transport, repairable, redundancy, hardening, availability, and physical defense. A complete explanation of the objectives and measures is presented in Appendix A Value Hierarchy. However, it should be noted that the evaluation measure ease of transport was integrated into the sub-objective mobile due to only having a single evaluation measure under mobile.

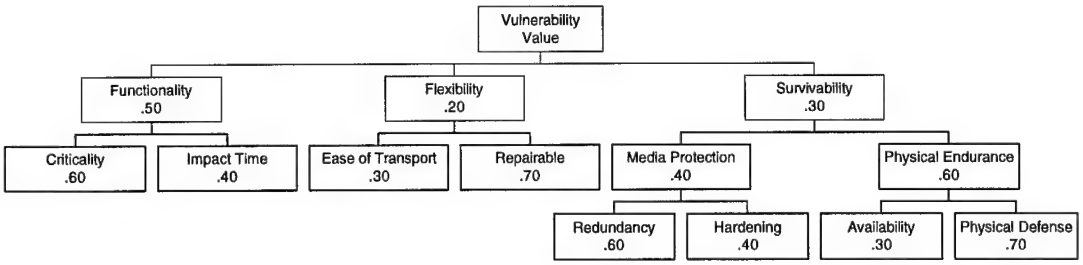


Figure 20 Value Hierarchy

The value model is used for scoring and ranking the vulnerability sets. The attributes of each node and link are used to derive the score of each vulnerability set. Having determined the vulnerability sets' scores, the multi-objective value function is used to evaluate each vulnerability set.

Given the weights and evaluation measures, using Keeney's additive utility function discussed in the literature search, and using Equation 3, the multi-objective vulnerability set value function is:

$$Value = \sum_{i=1}^{8} w_i V_i \left( \frac{\sum_{j=1}^{n} s_{ij}(x_{ij})}{n} \right) \qquad (4)$$

where $w_1$=.3, $w_2$=.2, $w_3$=.06, $w_4$=.14, $w_5$=.072, $w_6$=.048, $w_7$=.054, $w_8$=.126, and other terms are defined as in Equation 3.

The coefficients in the multi-objective value function are global weights. Global weights are calculated by multiplying the weight of the evaluation measure with the weights of objectives and sub-objectives which are above it in the hierarchy [6:557]. For example, the evaluation measure, criticality, has a local weight of 0.6, but a global weight of (0.6)*(0.5)=0.3, since the functionality objective has a local weight of 0.50. It is important to note that for evaluation purposes, the highest value any vulnerability set can obtain is one, and the lowest value is zero. This value range was determined from the single dimensional value functions discussed in Appendix A and the multi-objective vulnerability set value function shown above.

Table 5 Ranges for Evaluation Measures

| Evaluation Measure | Range(low, high) |
|---|---|
| Criticality<br>(average importance to network) | (3, 10) |
| Impact Time<br>(average time to impact reliability of network) | (0, 120) |
| Ease of Transport<br>(average transportability of components) | (2, 4) |
| Repairable<br>(average repair time of components) | (12, 96) |
| Redundancy<br>(average number of redundancies of a component) | (0, 2) |
| Hardening<br>(average number of hardening techniques of components) | (0, 1) |
| Availability<br>(average availability time of components) | (0.93, 0.99) |
| Physical Defense<br>(average number of defense techniques of components) | (1, 5) |

The range of scores for each evaluation measure are determined by averaging the component scores that comprise each vulnerability set and taking the high and low values for each evaluation measure. The ranges of component scores for each evaluation measure are shown in Table 5, while the ranges of scores of the vulnerability sets, resulting from analyzing the notional network, are shown in Table 6.

The values of the vulnerability sets are calculated, using the Visual Basic code shown in Appendix D, and the attributes of the nodes and links. The Visual Basic code uses Equations 1 and 2, Kirkwood's exponential value function equations. The value function domains used for each evaluation measure are the ranges of scores for each evaluation measure.

Each of the 34,285 vulnerability sets is evaluated against each evaluation measure and in accordance with the multi-objective value function, an overall assessment is

determined. The Visual Basic code used for evaluating the vulnerability sets and assessing the value is given in Appendix D Visual Basic Code.

Table 6 Ranges of Scores for Evaluation Measures

| Evaluation Measure | Range(low, high) |
|---|---|
| Criticality<br>(average importance to network) | (4.83, 9) |
| Impact Time<br>(average time to impact reliability of network) | (24, 96) |
| Ease of Transport<br>(average transportability of components) | (3, 4) |
| Repairable<br>(average repair time of components) | (15.43, 72) |
| Redundancy<br>(average number of redundancies of a component) | (0, 1.67) |
| Hardening<br>(average number of hardening techniques of components) | (0, 0.5) |
| Availability<br>(average availability time of components) | (0.94, 0.99) |
| Physical Defense<br>(average number of defense techniques of components) | (1, 3.33) |

All vulnerability sets are ranked using the vulnerability set value function. The ranked listing comprises the potential vulnerability sets for disrupting the network. Vulnerability sets 32750, 32752, and 32751 were ranked the highest with overall vulnerability set values of 0.87, 0.84, and 0.84 respectively, and vulnerability sets 33195, 33193, and 34069 had the lowest ranking with vulnerability set values of 0.30, 0.30, 0.29 respectively. Table 7 and Table 8 respectively show the highest and lowest ranking vulnerability sets along with their components.

Table 7 Highest Ranking Vulnerability Sets

| Vulnerability Set 32750 | Vulnerability Set 32752 | Vulnerability Set 32751 |
|---|---|---|
| Node 2: Socorro Optical Sensor<br>Node 3: Maui Optical Sensor<br>Node 4: Diego Garcia Optical Sensor<br>Value: 0.87 | Node 3: Maui Optical Sensor<br>Node 4: Diego Garcia Optical Sensor<br>Node 5: Primary Router<br>Value: 0.84 | Link 2-5: Primary Data Line<br>Node 3: Maui Optical Sensor<br>Node 4: Diego Garcia Optical Sensor<br>Value: 0.84 |

Table 8 Lowest Ranking Vulnerability Sets

| Vulnerability Set 33195 | Vulnerability Set 33193 | Vulnerability Set 34069 |
|---|---|---|
| Link 7-14: Satellite Relay<br>Link 7-18: Satellite Relay<br>Node 10: Tertiary Router<br>Link 13-16: Primary Data Line<br>Node 15: Secondary Router<br>Node 17: Tertiary Router<br>Value: 0.30 | Link 7-14: Satellite Relay<br>Link 7-18: Satellite Relay<br>Node 10: Tertiary Router<br>Node 13: Peterson AFB<br>Node 15: Secondary Router<br>Node 17: Tertiary Router<br>Value: 0.30 | Link 7-18: Satellite Relay<br>Node 10: Tertiary Router<br>Node 17: Tertiary Router<br>Node 19: Cheyenne Mountain AS<br>Value: 0.29 |

In order to analyze the vulnerability sets in more detail, a reduced list of the potential vulnerability sets, containing the twenty sets of highest value, is created. This reduced list is shown in Appendix C Notional Network Data.

Value Composition

Figure 21 graphically represents the values of the top twenty vulnerability sets according to how much each major objective (top level of the hierarchy) in the hierarchy contributes to the overall vulnerability set value. The major objectives are functionality, flexibility, and survivability. Appendix C Notional Network Data provides a table showing the numerical values of each objective for the top twenty vulnerability sets. Figure 21 shows that the contribution of functionality stands out as the most important impact, followed by survivability. This is expected due to the weights given to each objective, with functionality at .50, flexibility at .20, and survivability at .30.

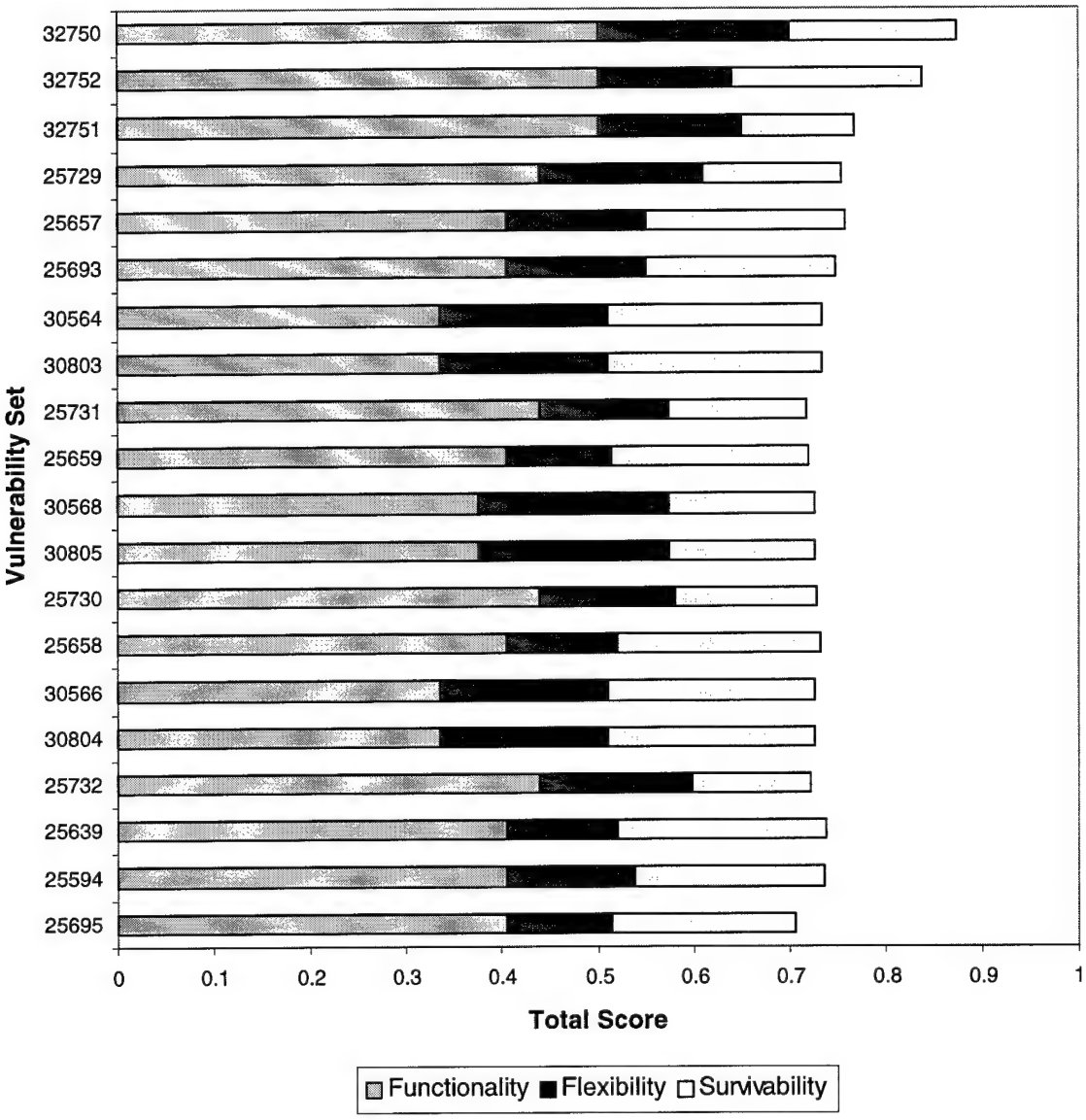


Figure 21 First Level Vulnerability Set Value Composition

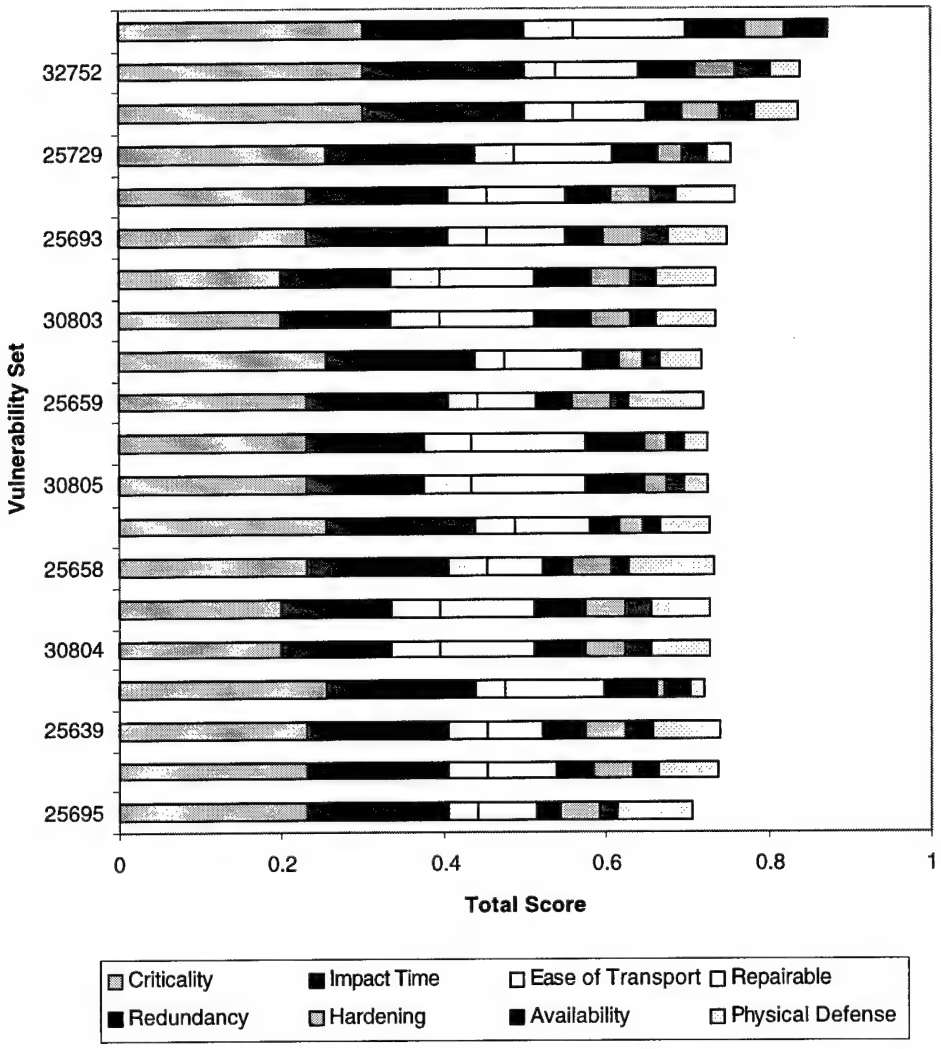


Figure 22 Evaluation Measure Vulnerability Set Value Composition

(Note: As you read the legend from left to right, top to bottom, the corresponding evaluation measure is shown left to right in the graph.)

Figure 22 graphically represents the values of the vulnerability sets according to how much each evaluation measure (lowest level of the hierarchy) contributes to the overall set value. Appendix C Notional Network Data provides a table showing the numerical values of each measure for the top twenty vulnerability sets.

From Figure 22, criticality stands out as the most important, followed by impact time. Again, this is expected due to the weights given to each measure and the weights of their respective major objective and sub-objectives.

One-way Sensitivity Analysis

In an effort to gain insight into the importance the weights have on the potential vulnerability sets, sensitivity analysis is conducted by swinging the weights for each evaluation measure. It is important to note that as an evaluation measure's weight approaches zero, that evaluation measure essentially becomes a non-player in the decision at hand. Similarly, a weight that approaches one makes the problem less of a decision analysis problem and more of an optimization effort for that evaluation measure. In this phase of the analysis, only the top five of the potential vulnerability sets are analyzed for sensitivity to changes in the objective and evaluation measure weights.

Figures 23 through 26 display the sensitivity of the five vulnerability set rankings to the weight of each objective and evaluation measure in the hierarchy. The base weights are shown as vertical broken lines. In addition, the figures display the points at which a vulnerability set drops below the top twenty ranked vulnerability sets. Appendix C Notional Network Data contains the data sets that are associated with each graph. A figure for each objective, sub-objective, and evaluation measure is not shown. This is due to the mirror image effects of evaluation measures or sub-objective under the same objective. For example, criticality is simply the mirror image of the impact time graph.

The figures show that the vulnerability sets are most sensitive to the major objectives of the hierarchy: functionality, flexibility, and survivability. As these weights are swung from 0 to 1 not only do the rankings of the top five vulnerability sets change, but most of the sets fall below the top twenty at some point. For example, in the case of survivability, set 25729 falls below the top twenty by increasing the weight from .3 to .5. In fact, the only time set 25729 falls within the top twenty sets is when the survivability weight is at a weight of .4 or below. Similarly, the ranking of the top five vulnerability sets changes in all three cases with a small decrease or increase in the weights.

The evaluation measure ease of transport is the only other graph that shows a vulnerability set dropping below the top twenty. However, the point where the set drops below the top twenty is six weight shifts away from the base value, which is not very likely to occur. As seen in Figure 26, Availability (and its complement physical defense) weight makes very little to the sets due to the fact that there is very little variation in scores for those attributes. Thus a variation in the weights associated with these evaluation measures would make no difference to the ranking of those sets. However, redundancy and its mirror image evaluation measure, hardening, have their base weights on a point at which a shift higher or lower would change the vulnerability set ranking. Therefore a variation in weight for these measures does make a difference to the overall ranking of the vulnerability sets. However, the top vulnerability set is unaffected by the variation in weights. The scores are shown in Appendix C Notional Network Data.

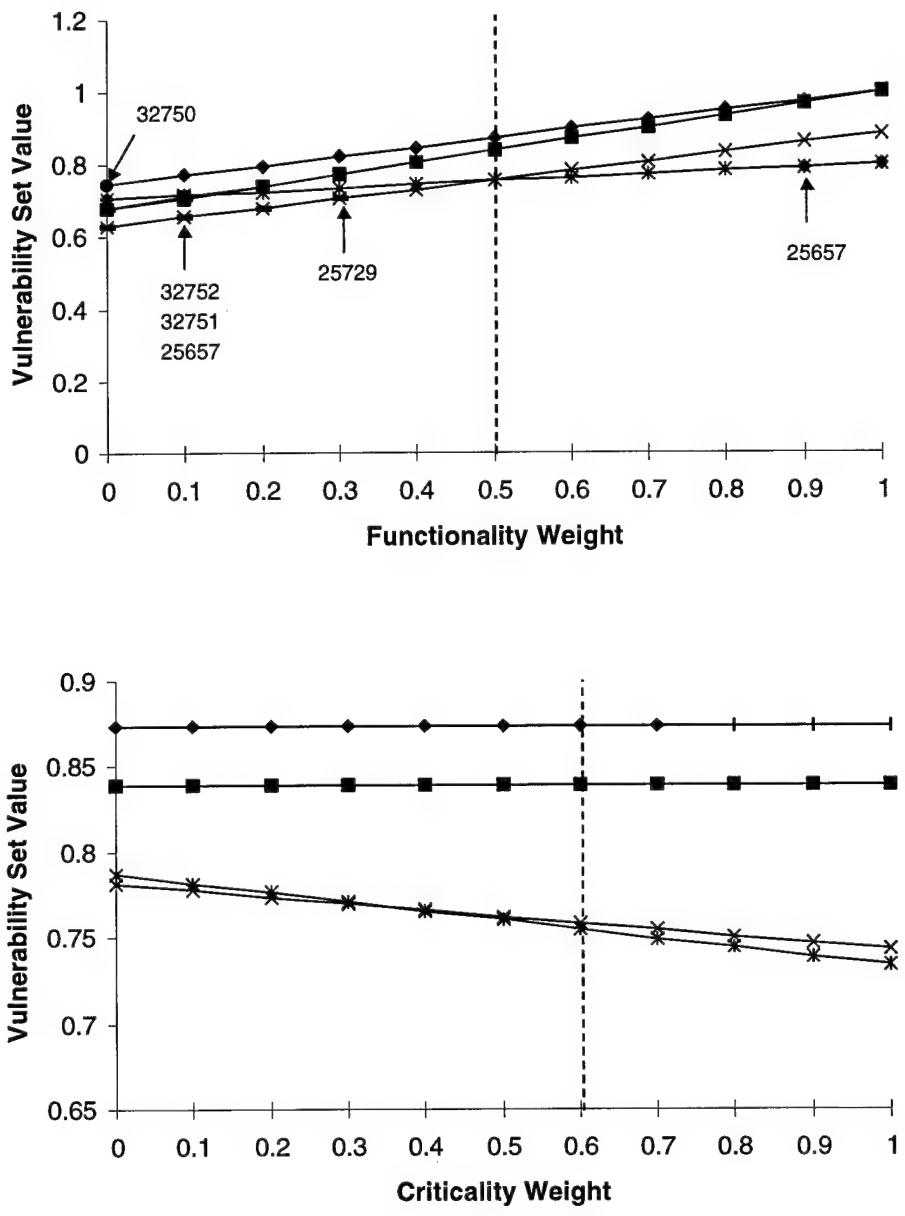


Figure 23 Functionality and Criticality Sensitivity Analysis

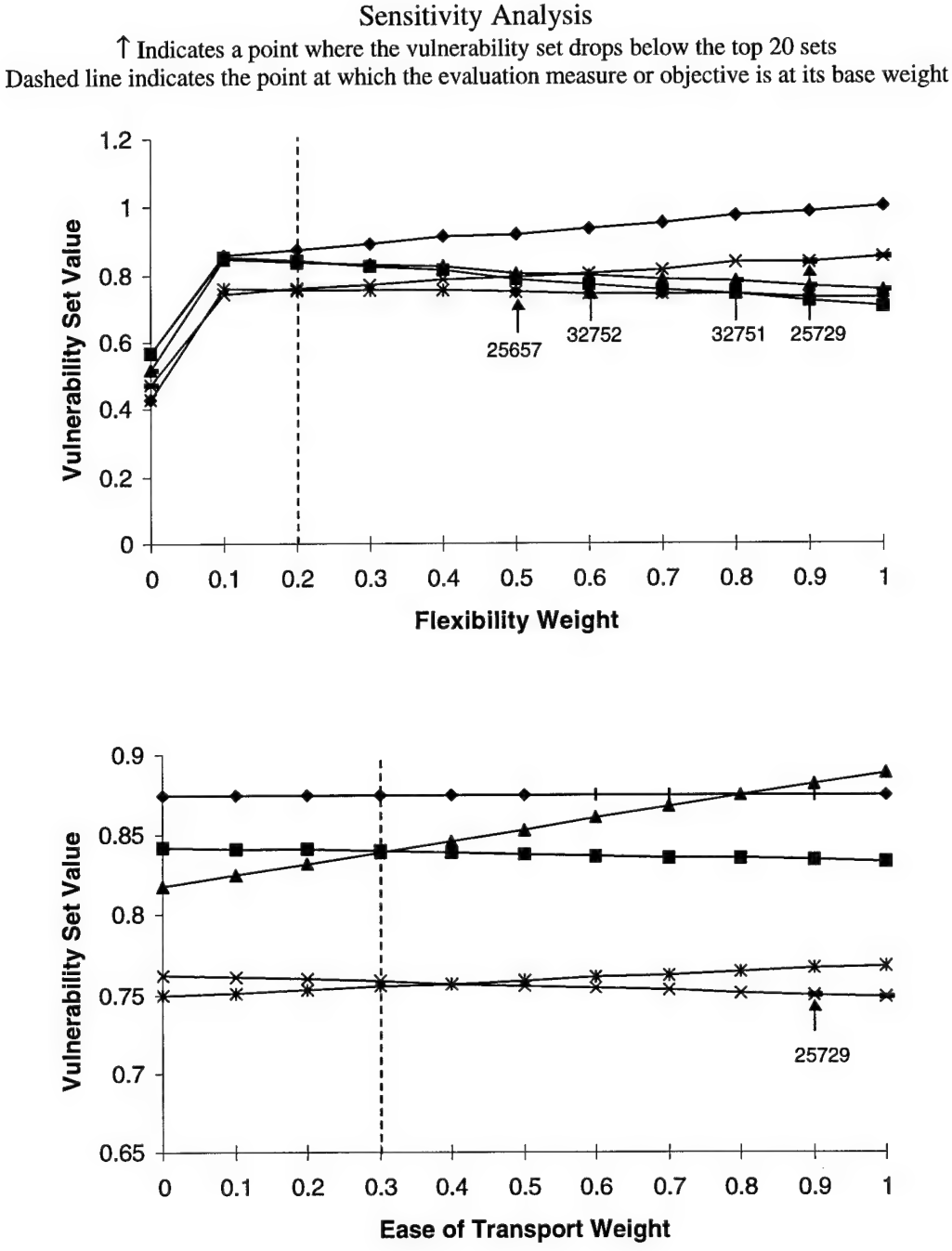


Figure 24 Flexibility and Ease of Transport Sensitivity Analysis

Sensitivity Analysis

↑ Indicates a point where the vulnerability set drops below the top 20 sets

Dashed line indicates the point at which the evaluation measure or objective is at its base weight

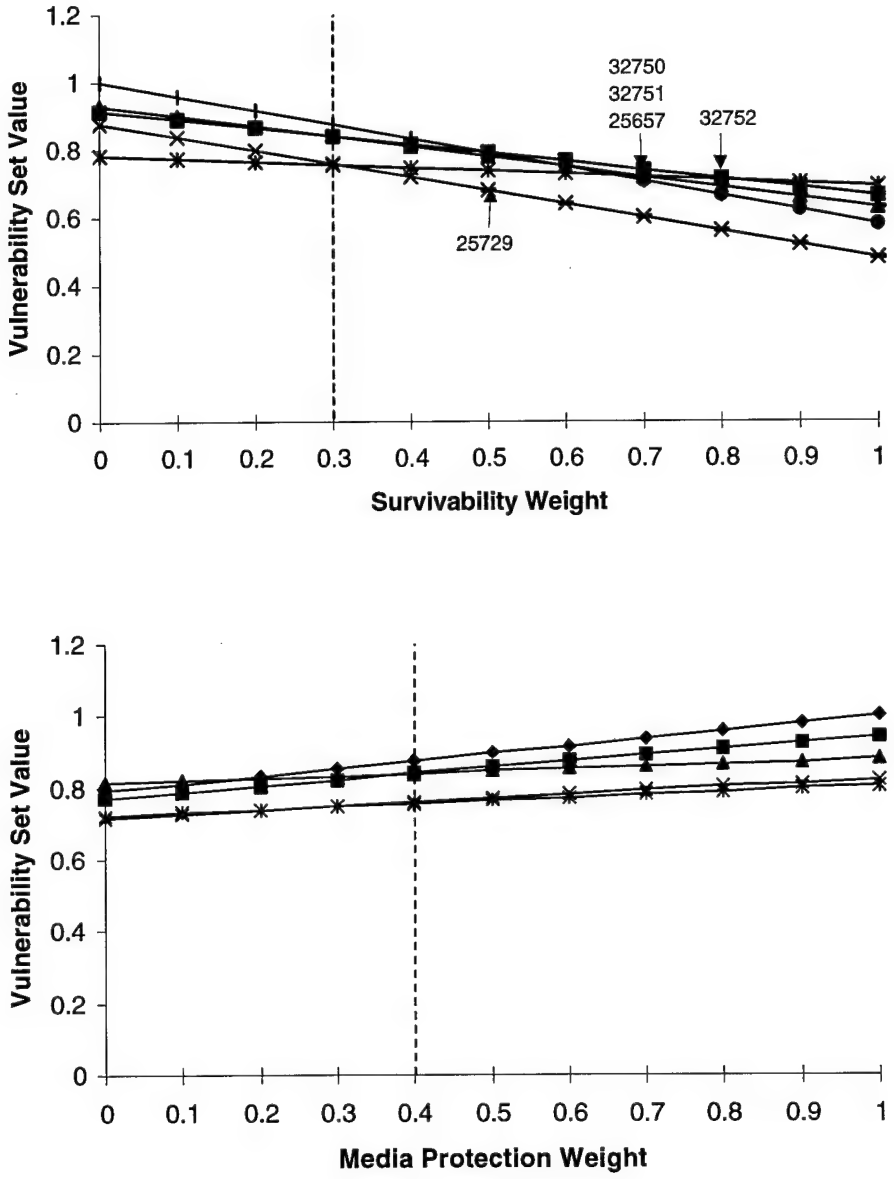


Figure 25 Survivability and Media Protection Sensitivity Analysis

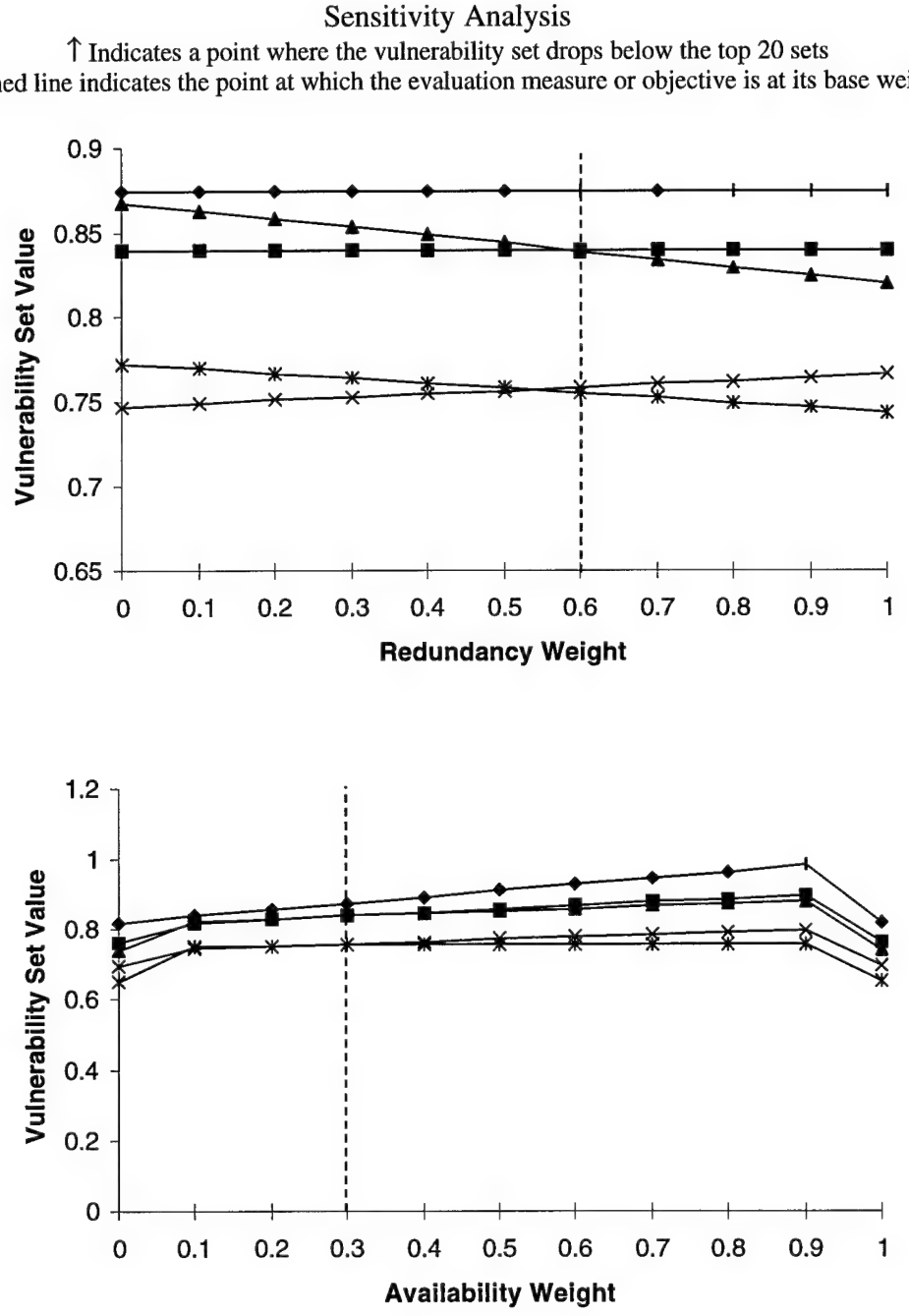


Figure 26 Redundancy and Availability Sensitivity Analysis

Figures 23 and 25 show that the evaluation measures criticality and impact time and the sub-objectives media protection and physical endurance are not sensitive to the weights being varied. The 4th and 5th ranked sets do switch rankings in criticality and its complement, impact time, just beyond the base weights. However, none of the sets fall below the top twenty. Media protection and its complement, physical endurance, cause only the third ranked vulnerability set to the change as the weight changes. Therefore, further study should be done to verify these weights and determine which direction they might move.

Availability and its complement, physical defense, are the least sensitive evaluation measures. The vulnerability sets always remain in the top twenty. Neither measure causes the ranking of the vulnerability sets to change.

The sensitivity graphs give insight into how sensitive the vulnerability sets and their rankings are to changes in the objectives and evaluation measures' weighting. Further research into the certainty of the most sensitive evaluation measure's weight is certainly required in order to make the proper vulnerability set selection. A proper selection and prioritization of the vulnerability sets would allow a decision maker the opportunity to increase survivability, security, and maintain throughput of the network. In addition, due to possible budget or time constraints the entire set of vulnerabilities may not be examined. Thus a prioritized list is needed.

Persistency Analysis

Additional analysis included component recurrence, or persistence. Leinart presented recurrence or persistence as how often the component is found in the top twenty vulnerability sets at the base weight [20:4-22]. Here persistency analysis is defined to be how often the component is found in the top twenty vulnerability sets as the weight of the evaluation measure or objective is swung from 0 to 1. Figures 27 through 30 show graphs of persistence for all the nodes and links in the top twenty vulnerability sets as the weights of the evaluation measures or objectives were swung from 0 to 1.

Again, functionality, flexibility, and survivability are the most sensitive to the change in weights. Not only does the number of times a node or link occurs in the top twenty change, but the total number of nodes and links change as the weight changes. As shown in Figure 29, media protection (and its complement, physical endurance) is sensitive to the change in weights. The base weight for media protection would have to increase to .8 and physical endurance drop to .2 before stabilization occurs. However, no new nodes or links are introduced in either sub-objective. Figure 30 shows availability and its complement, physical defense, are most sensitive as availability decreases and physical defense increases. Ease of transport and its complement, repairable, are very sensitive as ease of transport increases to approximately .6 and repairable decreases to .4, as seen in Figure 28.

Persistency Analysis

Dashed line indicates the point at which the evaluation measure or objective is at its base weight

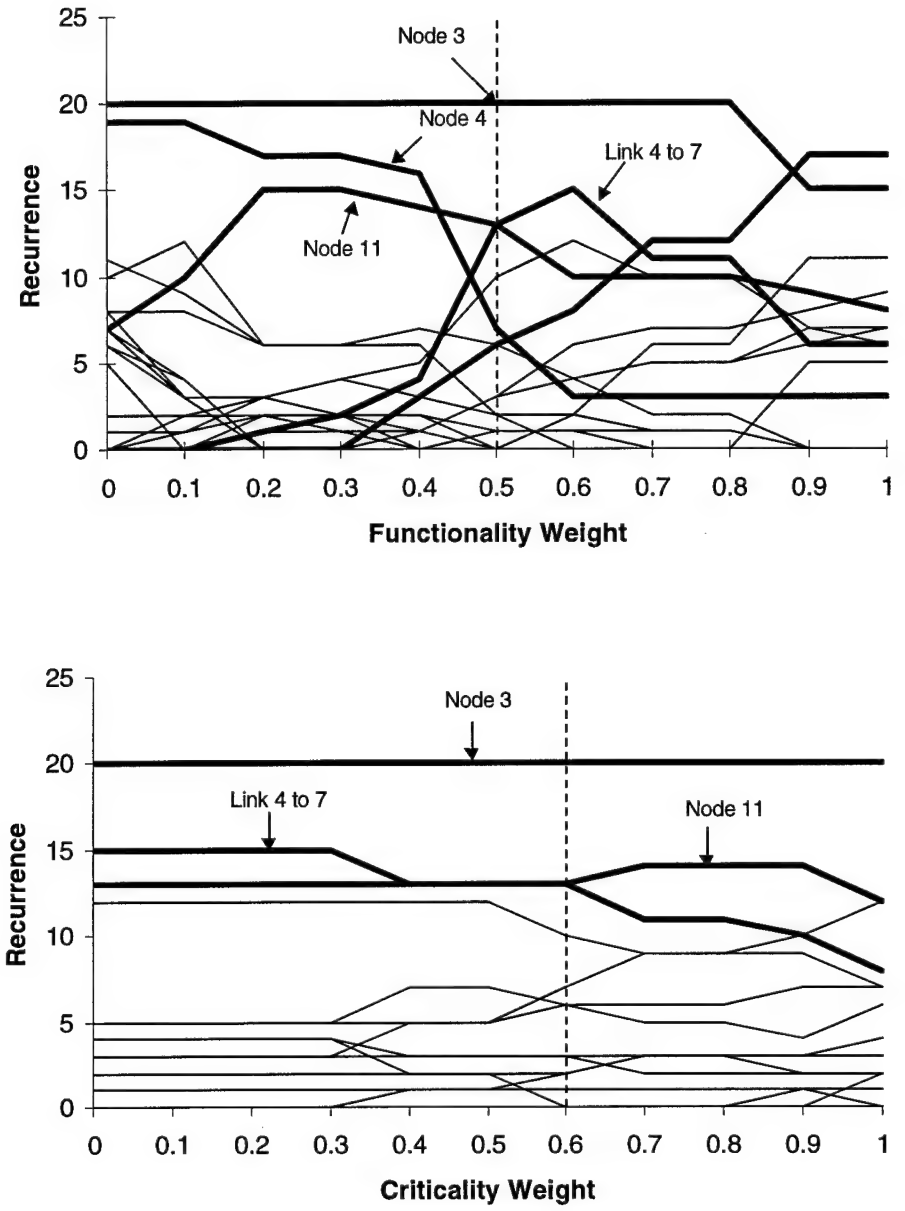


Figure 27 Functionality and Criticality Persistency Analysis

Persistency Analysis

Dashed line indicates the point at which the evaluation measure or objective is at its base weight

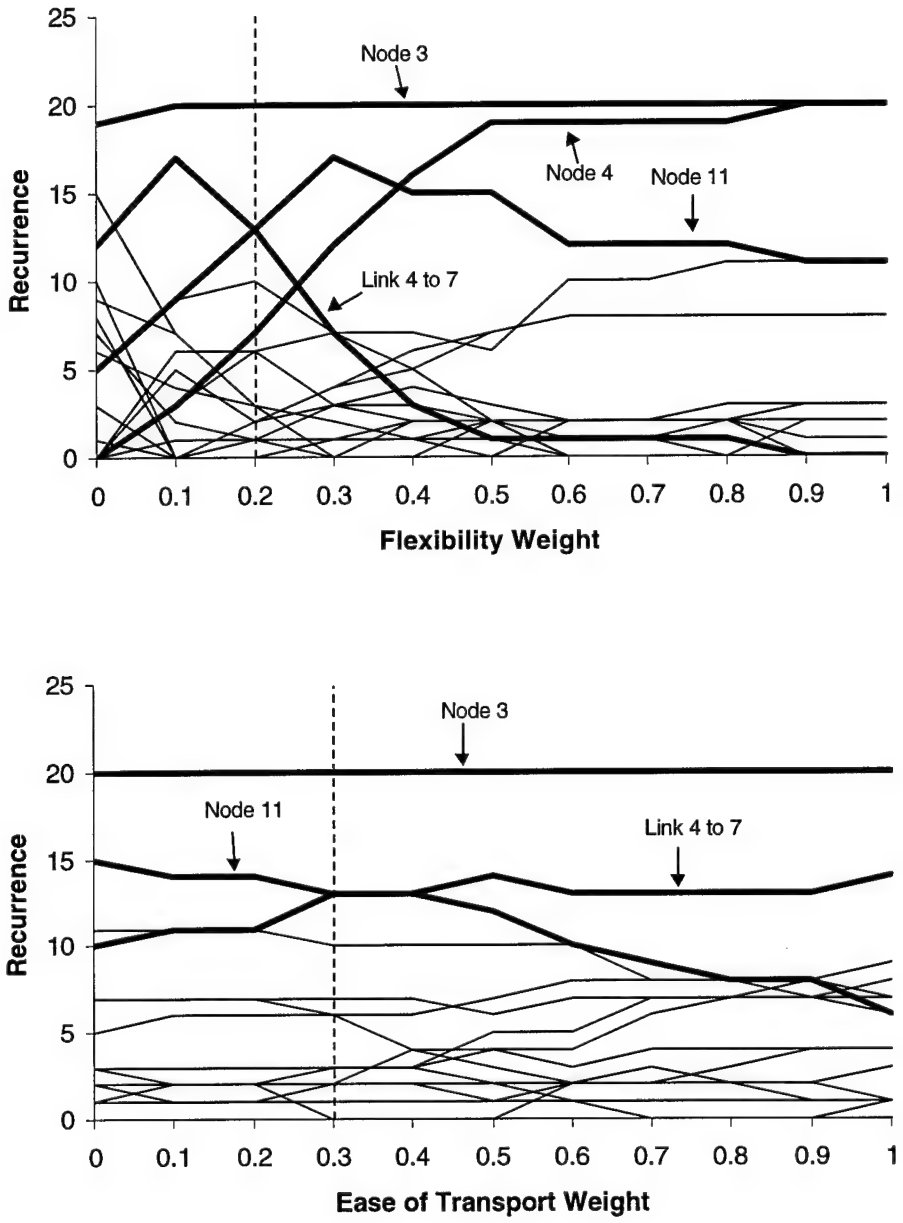


Figure 28 Flexibility and Ease of Transport Persistency Analysis

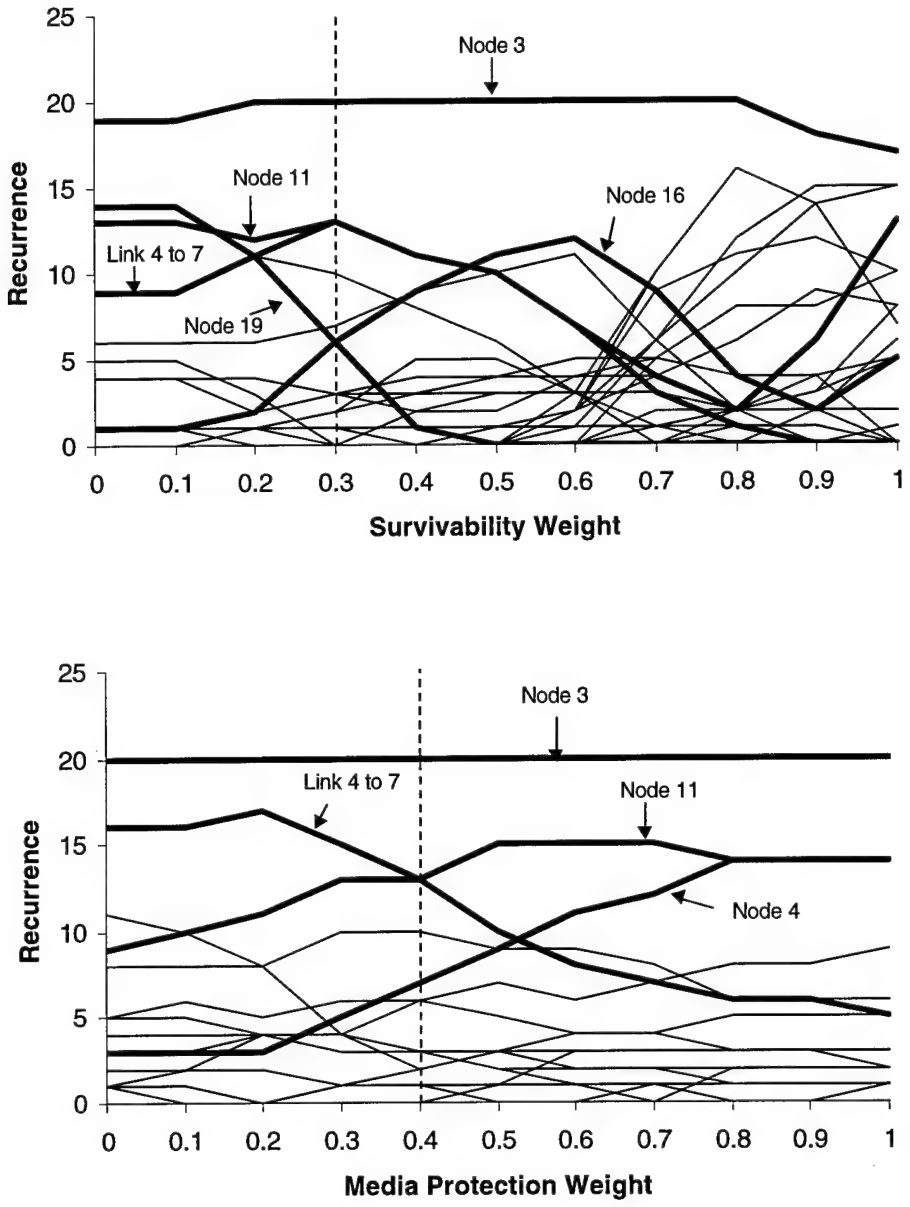


Figure 29 Survivability and Media Protection Persistency Analysis

Persistency Analysis

Dashed line indicates the point at which the evaluation measure or objective is at its base weight

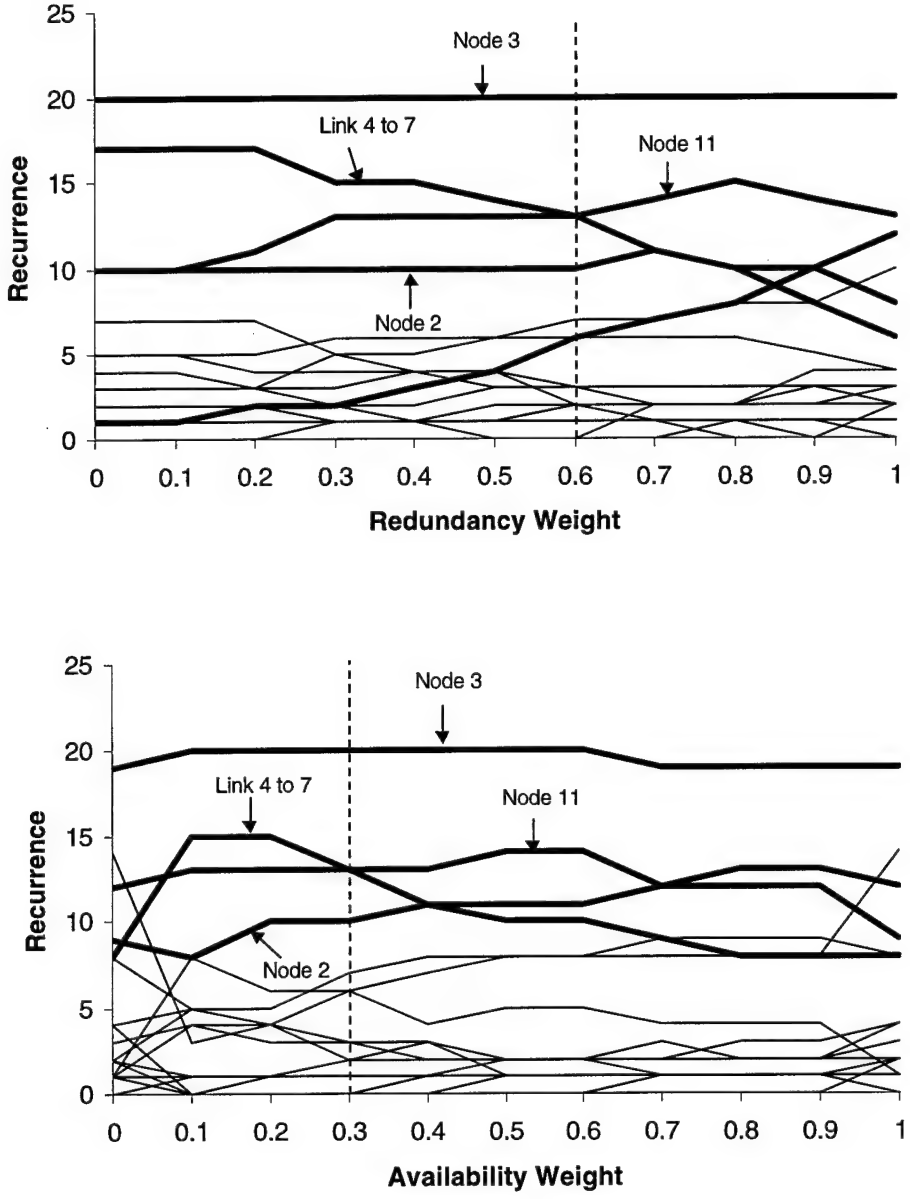


Figure 30 Redundancy and Availability Persistency Analysis

The evaluation measures criticality and impact time are the least sensitive in the persistency analysis to the change in weights. There is small variation in the persistency; however, no new nodes or links are introduced into the top twenty in either case. It is also important to note that Node 3 remains the most persistent component throughout the analysis except in the functionality objective, where it drops as the weight is increased to .8, which is unlikely since this is several increments from the base weight. Clearly, Node 3 is an important component of the network that may require further analysis.

Figure 31 shows the percent occurrence of each node and link in all vulnerability sets. For example, Node 2 occurs in vulnerability sets approximately 19% of the time. In other words, 19% of all 34,285 vulnerability sets will contain Node 2. As shown in Figure 31, Node 15 occurs in approximately 27% of the sets and Link 7 to 18 occurs in approximately 50% of the sets, and are the most recurring node and link respectively. In addition, Node 12 and Node 17 are closely behind at 24%, as well as Link 5 to 13 and Link 6 to 13 at 35%. These nodes and links could pose a potential threat due to the number of times it occurs in a vulnerability set. Thus further analysis of these nodes and links is suggested.

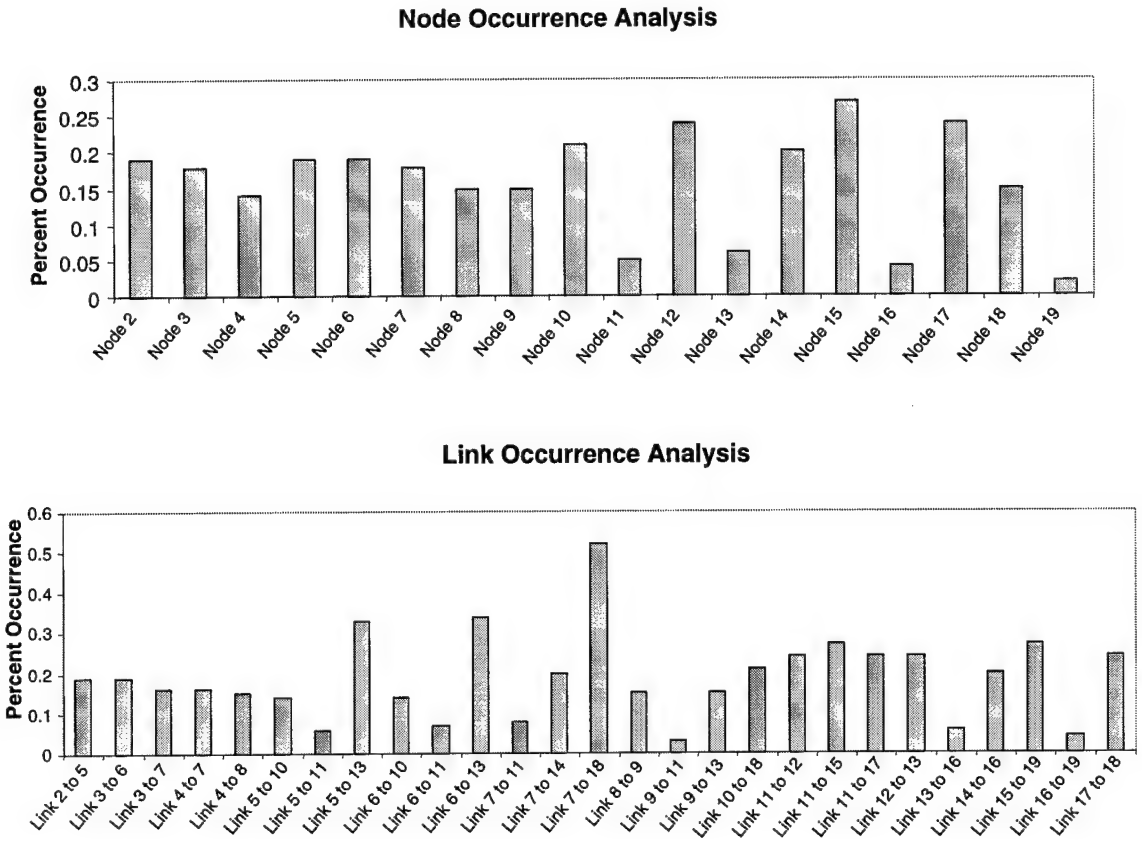


Figure 31 Node and Link Occurrence Charts

Sub-Network Analysis

As was stated earlier, Nodes 2, 3, 4, 11, 18, and 19 were used to perform sub-network analysis. The sub-network components represent the power and water resources necessary to operate the respective sites. The components and attribute data are listed in Appendix C Notional Network Data.

Using the same value hierarchy and the sub-network attribute data, shown in Appendix C, a ranked listing of sub-network vulnerabilities is assessed. Each component of the sub-network is scored and evaluated using the appropriate value function. Using

Equation 4 the value can be assessed. However, in this case n=1 and there is no need to average individual component scores. Instead the score of the component is directly input into the value function.

Table 9 shows the ranked list of sub-network components for each node analyzed. Using the hypothetical value hierarchy, we can conclude that the most effective way to take out Node 2 is to destroy the node itself. This is a result of the impact time and criticality of the component. In addition, the power station, which is the next ranked vulnerability for Node 2, is shown to have a redundancy in the attribute data (see Appendix C). The node is determined to have a backup power generator. In the case of Node 18, the most effective way to disrupt the node is to destroy the water station. This is due to the low value for the criticality evaluation measure and the fact that both the node itself and the power resources included a redundancy.

Using the information in Table 9 and the previous persistency data, one can conclude that Node 3 is the most vulnerable node and the most effective way to disrupt that node is to destroy the node itself or its power station. This data can be used to identify individual vulnerabilities, thus allowing a decision maker to focus in on the most vulnerable components of the network or make informed decisions considering susceptibility. This focus allows the decision maker to make protection plans and expend the budget in a prioritized manner.

Table 9 Sub-Network Vulnerability Ranking

| Components | Description | Value |
|---|---|---|
| ***Node 2*** | ***Socorro Optical Sensor*** | ***0.73*** |
| 2P1 | Socorro Power Station | 0.56 |
| Link 2P1 to 2 | Socorro Power Line | 0.55 |
| 2W1 | Socorro Water Station | 0.36 |
| 2W2 | Socorro Water Station | 0.35 |
| Link 2W1 to 2W2 | Socorro Water Line | 0.28 |
| Link 2W2 to 2 | Socorro Water Line | 0.28 |
| ***Node 3*** | ***Maui Optical Sensor*** | ***0.83*** |
| 3P1 | Maui Power Station | 0.73 |
| Link 3P1 to 3P2 | Maui Power Line | 0.65 |
| 3P2 | Maui Power Station | 0.64 |
| Link 3P2 to 3 | Maui Power Line | 0.58 |
| ***Node 4*** | ***Diego Garcia Optical Sensor*** | ***0.72*** |
| 4P1 | Diego Garcia Power Station | 0.64 |
| Link 4P1 to 4 | Diego Garcia Power Line | 0.62 |
| 4W1 | Diego Garcia Water Station | 0.32 |
| Link 4W1 to 4 | Diego Garcia Water Line | 0.28 |
| ***Node 11*** | ***OC3F – Edwards AFB*** | ***0.68*** |
| Link 11P1 to 11 | Edwards AFB Power Line | 0.62 |
| 11P1 | Edwards AFB Power Station | 0.60 |
| ***Node 18*** | ***NAVSPACE*** | ***0.28*** |
| 18W1 | NAVSPACE Water Station | 0.35 |
| Link 18W1 to 18 | NAVSPACE Water Line | 0.32 |
| Link 18P1 to 18 | NAVSPACE Power Line | 0.31 |
| 18P1 | NAVSPACE Power Station | 0.29 |
| ***Node 19*** | ***SCC - Cheyenne Mountain Air Station (CMAS)*** | ***0.74*** |
| 19W1 | Colorado Springs Water Station | 0.70 |
| 19P3 | Colorado Springs Power Station | 0.66 |
| 19P2 | Colorado Springs Power Station | 0.65 |
| Link 19P2 to 19P3 | Colorado Springs Power Line | 0.61 |
| Link 19W1 to 19W2 | Colorado Springs Water Line | 0.61 |
| Link 19P3 to 19 | CMAS Power Line | 0.59 |
| 19W2 | CMAS Water Station | 0.57 |
| 19P1 | CMAS Power Generator | 0.55 |
| Link 19P1 to 19 | CMAS Power Line | 0.55 |
| Link 19W2 to 19 | CMAS Water Line | 0.50 |

## Summary

Given the hypothetical network of GEODSS sensors, a list of potential vulnerability sets was determined. In addition, a final list of the highest ranking

vulnerability sets was analyzed using a hypothetical value model established from a combination of Space Command resources discussed in the methodology chapter.

Sensitivity analysis was then conducted on the five vulnerability sets with the highest vulnerability value. This sensitivity analysis identified functionality, flexibility, survivability, repairable, and ease of transport as critical in assessing the rankings of the vulnerability sets. Further study in the uncertainty of the weights for these objectives and evaluation measures should be conducted.

Persistency analysis suggested that the sub-objectives media protection and physical endurance should be included in further studies. This analysis focused on individual components instead of the vulnerability sets. In addition, the occurrence data provided a means for determining which nodes and links occurred more often in the vulnerability sets.

Sub-network analysis was then conducted on several nodes to show how an individual node could be disrupted by other means. The analysis provided a ranked list of sub-network components that could disrupt the entire network by disrupting the associated node. Again, changes in the value model could change the outcome of the rankings. Thus it is very important to establish a valid working value model.

# V. Conclusions and Recommendations

This chapter provides conclusions from the analysis completed and makes recommendations for further related research.

## Overview

This effort accomplished the goal of providing a tool and methodology for generating vulnerabilities of a network and a quantitative method to determine the value for ranking the vulnerabilities. The ranking achieves the overall objective of finding the most vulnerable components of a network which could reduce the reliability of the network from one to zero as the number of paths in the network goes from one or more to zero. In addition, the methodology allows sub-network vulnerabilities, such as water or power lines, to be identified. The Network Vulnerability Assessment Tool assists in implementing the methodology for evaluating a network. Additionally, the tool provides a means to graphically display and provide insight into the driving factors behind the scores used for ranking the vulnerabilities. This can assist the decision maker in determining the appropriate value weights.

## Research Results

The analysis conducted on the notional network was used to illustrate the concepts of generating network vulnerability sets, measuring these sets to nominate the most vulnerable set, evaluating the driving factors, and analyzing sub-network vulnerabilities. The methodology allows the tool to be applied to a variety of network

types and sub-networks. The tool allows for the value structure to be adjusted to reflect the realities of an actual network and its decision maker's values.

The value model presents an examination of the important factors involved in the decision of ranking the vulnerabilities based on knowledge, experience, and objectives of the decision maker. As the notional network showed, the method contributes a means to evaluate any network and its sub-network components. The insights gained from the sensitivity, persistency, and occurrence analysis are valuable to the decision maker.

The insights a decision maker gains from the assessment tool can be used to make protection plans that would increase survivability, security, and maintain throughput of the analyzed network. In addition, the prioritization of the vulnerabilities allows a decision maker to justify and prioritize the expenditure of limited budget resources.

## Limitations of the Study

The computational rate of the vulnerability set generation algorithm currently limits the application of the Network Vulnerability Assessment Tool to small networks. The value model input currently allows only a four level hierarchy to be input; however, refinements to the code would allow the user to input a larger hierarchy. In addition, the tool assumes deterministic network features.

## Recommendations for Future Research

As mentioned above, the computational rate of the vulnerability set (cut-set) generation algorithm is a limiting factor for large network analysis. Future research

should focus on developing a cut-set algorithm or generating computer code for an existing cut-set algorithm that would improve the computational rate.

Additional input into the value model from a decision maker would allow analysis of an actual network scenario. Furthermore, elicitation of the exact shape and ranges of the value functions would increase viability. The notional network was assumed to have linear value functions. This linearity was used to prove the methodology; however, in most network scenarios the value functions may have some point of diminishing returns implying non-linearities. Further analysis into the correct or most appropriate method of evaluating the value of an alternative with more than one component is suggested.

A graphical user interface and user-friendly macros would allow for wider dissemination and usability of the assessment tool.

Analysis using a real-world network would provide valuable insight into the assessment tool. In addition, it would identify any limitations the assessment tool would have in real-world scenarios.

**Conclusions**

The approach developed in this effort serves as a method for finding network vulnerabilities that have the highest value for disruption. Using expert opinion, the value model provides a benchmark and metric for ranking the vulnerabilities. The Network Vulnerability Assessment tool was applied to a notional network demonstrated actual application of the models. The outputs of the model, while not computationally quick, provide the decision maker with valuable information on components which could pose

as a possible threat for reducing the reliability of a critical network. This information can be used to make protection plans for the network and maintain control. Additionally, indirect vulnerabilities of sub-network components, which are not rapidly identified without the tool, can be identified. The concepts presented are proven and could be implemented in an operational environment.

# Bibliography


1 Bailey, Thomas G. Response Surface Analysis of Stochastic Network Performance. MS Thesis, AFIT/GOR/ENS-88D.

2 Bulteau, S. and G. Rubino. "A new approach to vulnerability evaluation of communication networks," Proceedings of the 15th International Teletraffic Congress: 681-690 (22-27 June, 1977)

3 Busacker, Robert G. and Saaty, Thomas L. Finite Graphs and Networks: An Introduction with Applications. New York: McGraw-Hill , Inc., 1965.

4 CACI Products Company. "COMNET III Product Description." Excerpt from unpublished article, n. pag. http://www.caciasl.com/COMNET_quick_look.html 1 October 1998.

5 California Scientific Software, Brain Maker, http://www.calsci.com/index.html. 1 October 1998

6 Clemen, Robert T. Making Hard Decisions: An Introduction to Decision Analysis (Second Edition). Pacific Grove CA: Brooks/Cole Publishing Company, 1996.

7 Davis, Christine C. A Methodology for Evaluating and Enhancing C4I Networks. MS Thesis, AFIT/ENS/GOR-97M

8 Department of the Air Force. AF2025 1998.

9 Department of the Air Force. Long Range Plan. United States Space Command. (1998). Available Internet: http://www.spacecom.af.mil/usspace

10 Department of the Air Force. Space Handbook: A War Fighter's Guide to Space. AU-18. Maxwell AFB, Air University Press, 1993.

11 Department of the Air Force. Space Operations. Air Force Doctrine 2-2. (1998)

12 Engel, Thomas G. "Splice: A new Analytical Network Analysis Software." Excerpt from unpublished article, http://fairway.ecn.purdue.edu/v1/asee/fie95/2c6/2c65/2c65.htm. 1 October 1998.

13 European Circuit Society, ESA Circuit Analysis Program, http://www.it.dtu.dk/~el/ecs/esacap.htm. 1 October 1998.

14 Hao, Jianxiu. "A Faster Algorithm for Finding the Minimum Cut in a Directed Graph," Journal of Algorithms. Vol 17: 424-446 (1994)

15 ISE1 group project. Excerpts from an unpublished article, n. pag http://www.doc.ic.ac.uk/~ih/ise1/projects/nwtool/97/analysis.html 19 October, 1998.

16 Keeney, Ralph L. Value-Focused Thinking. Cambridge MA: Harvard University Press, 1998.

17 Kirkwood, Craig W. Strategic Decision Making: Multiobjective Decision Analysis with Spreadsheets. Belmont: Duxbury Press, 1997.

18 Koester, David P. (1995). "Node-tearing Nodal Analysis." Excerpt from unpublished article, n. pag. http://www.npac.syr.edu/techreports/hypertext/sccs-6/9/node23.html 2 July 1998.

19 KrackPlot, http://www.contrib.andrew.cmu.edu/~krack/index.html. 1 October 1998

20 Leinart, James A. A Network Disruption Modeling Tool. MS Thesis, AFIT/GOR/ENS/98M-15.

21 Lithec, ModelQuest. http://www.lithec.sk/uk/dts_modelquest.htm. 1 October 1998.

22 Liu, Chen-Ching. "Analysis of Vulnerability of Power Network Configurations," Proceedings of ISCAS 85. 1513-1515. IEEE Press, 1985.

23 Mil3, OPNET, http://www.mil3.com/products/home.html. 1 October 1998

24 NetSense Internet Solutions. http://www.netsense.net/. 1 October 1998.

25 Novell, LANalyzer. http://137.65.2.6/catalog/qr/sne54200.html. 1 October 1998.

26 ODS Networks, Inc. http://www.ods.com. 1 October 1998.

27 Patvardhan, C., Prasad, V. C. and V. Prem Pyara. "Vertex Cutsets of Undirected Graphs," IEEE Transactions on Reliability. Vol 44, No. 2: 347-353 (June 1995).

28 Provan, J. S. and D. R. Shier. "A Paradigm for Listing (s,t)-Cuts in Graphs," Algorithmica. Vol 15: 351-372 (1996).

29 Shier, D.R., and D.E. Whited. "Algorithms for Generating Minimal Cutsets by Inversion," IEEE Transactions on Reliability. Vol 34, No. 4: 314-319 (October 1985)

30 Tahoe Design Software. http://www.ltol.com~tds/index.html. 1 October 1998.

31 The American Heritage Dictionary. New York NY: Dell Publishing, 1989.

32 Wilson, Robin J. Introduction to Graph Theory (Fourth Edition). England: Longman Group Ltd, 1996.

# Appendix A Value Hierarchy

This appendix describes the value hierarchy constructed to evaluate the network vulnerability sets. Each measure, the value function, and weights are presented and explained.

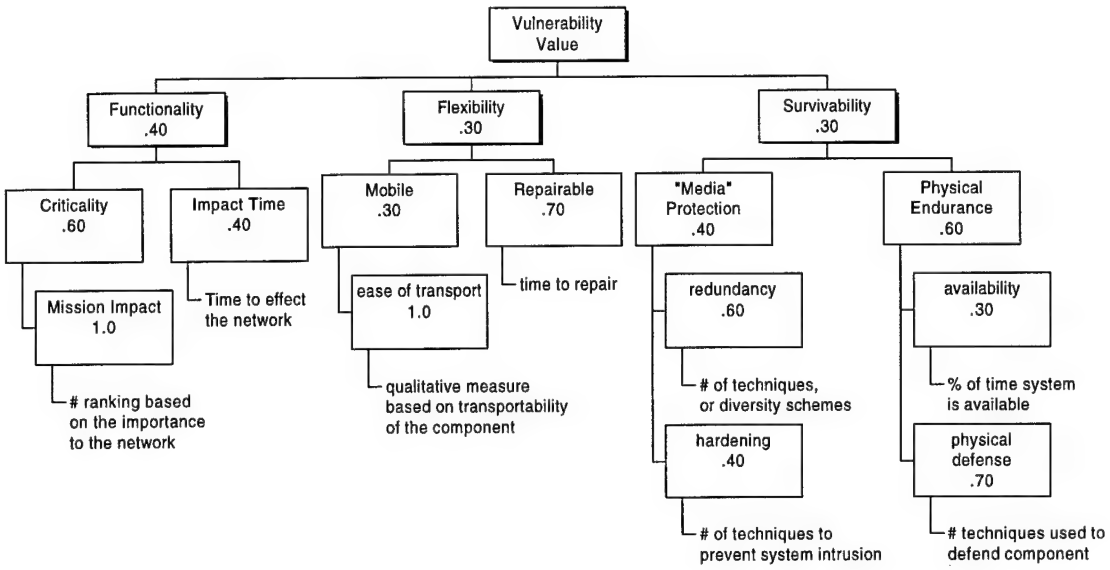


Figure 32 Vulnerability Value Hierarchy with weights

The overall goal is to rank vulnerability sets. This is broken down into three key objectives defined as follows:

1. *Functionality.* The reactions and impacts of the network in a stressed environment.
2. *Flexibility.* The ability to adapt to a stressed environment.
3. *Survivability.* The ability to perform in a stressed environment.

Functionality

This key objective consists of two evaluation measures: *criticality* and *impact time*. An assessment of the impact to the system in a stressed environment and the importance of that impact are critical when trying to find the most vulnerable components.

*Criticality*: This measure assesses the mission impact. It determines the importance of the component to the network. The more important the component, the higher the vulnerability of the network if the component is disrupted. If the component is critical to the network it is assigned a value of 10. If the component has no impact to the network it is assigned a value of 0. The single dimensional value function is shown in Figure 33.

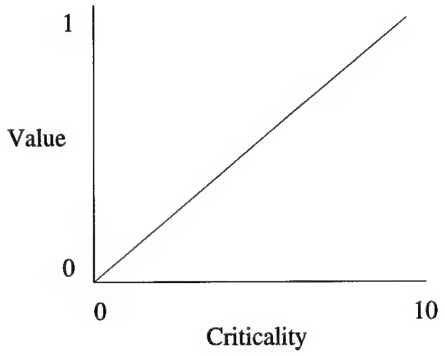


Figure 33 Criticality Value Function

*Impact Time*: This measure estimates the time it will take, after disruption, for the network to be impacted. The less time it takes for a disrupted component to impact the network, the more vulnerable it is to the system. *Impact Time* is measured in hours. The

component with longest time is assigned a value of one. The value function is shown in Figure 34.

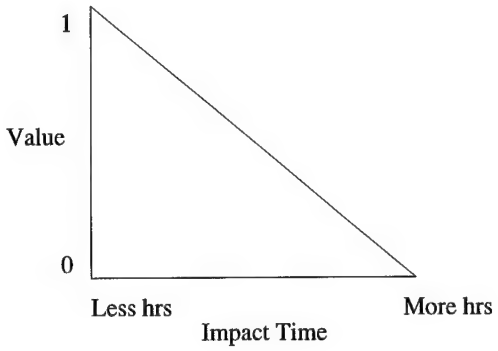


Figure 34 Impact Time Value Function

Flexibility

This key objective is decomposed into a sub-goal and a measure: *mobile* and *repairable*. They capture the importance of flexibility in a network. Flexibility provides the capability to adapt to different environments. The harder it is to adapt, the more vulnerable the network becomes.

*Mobile*: This sub-goal is defined by ease of transport. If a component cannot be transported then it is easier to target, thus it becomes more vulnerable. As Davis stated, "this is a qualitative measure based on the transportability of the system. The scores are based on four categories: manpacks, vehicle mounted, fixed transportable, fixed permanent" [7]. Figure 35 shows the mobile value function represented as ease of transport.

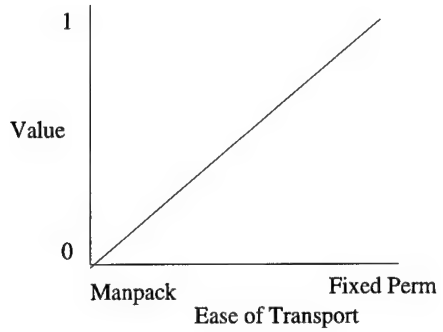


Figure 35 Ease of Transport Value Function

*Repairable*: This measure determines how fast a component can be repaired in a stressed environment. A component that is quicker to repair is less vulnerable to the network. The time to repair is the number of hours it takes to repair the disrupted component until the component no longer impacts the network. The value function is shown in Figure 36.

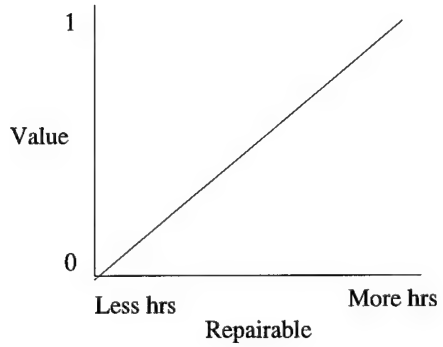


Figure 36 Repairable Value Function

Survivability

Survivability includes both the protection of the media within the network (media protection) and the physical protection of the network components (physical endurance).

In order to determine which components are more vulnerable, an assessment on how they can survive in a stressed environment must be performed.

*Media Protection*: This sub-goal consists of two measures: *redundancy* and *hardening*. These measures determine how well the media in the network is protected.

*Redundancy* is used to measure the number of techniques used to provide redundancy in the network. The more redundancy the less vulnerable the component becomes. Figure 37 shows the value function.

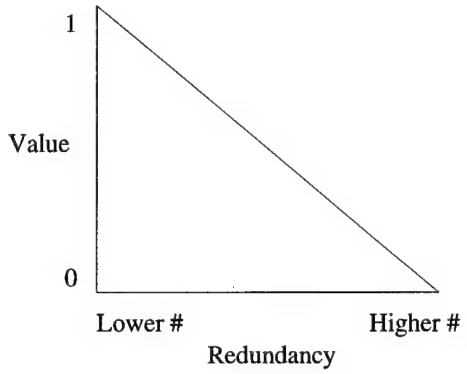


Figure 37 Redundancy Value Function

*Hardening* is used to measure the number of techniques used to prevent intrusion into the system. It indicates how well the media is protected from intruders. The value function is shown in Figure 38.

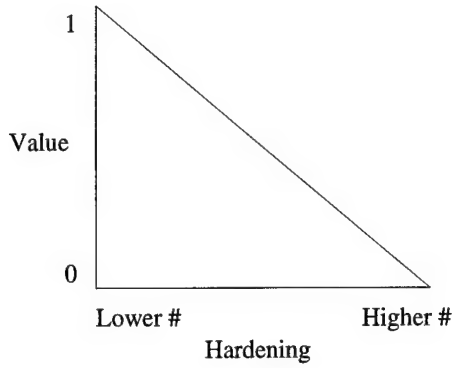


Figure 38 Hardening Value Function

*Physical Endurance*: This sub-goal includes two measures: *availability* and *physical defense*. These measures determine how well the components of a system perform and how well they are protected.

*Availability* is used to describe the availability of a component. Availability was defined by Davis, "as the percentage of time the component is available to be used by the network" [7]. It is expressed as

$$\text{Availability} = (\text{MTBF})/(\text{MTBF} + \text{MTTR})$$

where MTBF is mean time between failures and MTTR is mean time to repair a failure. Availability can then be used to calculate the total availability of a system comprised of components arranged in parallel and series structures. If n components are in parallel

$$A = 1 - (\Pi_{i=1 \text{ to } n}(1-A_i))$$

If n components are in series

$$A = \Pi_{i=1 \text{ to } n}(A_i)$$

[7:B-15] The value function is shown in Figure 39.

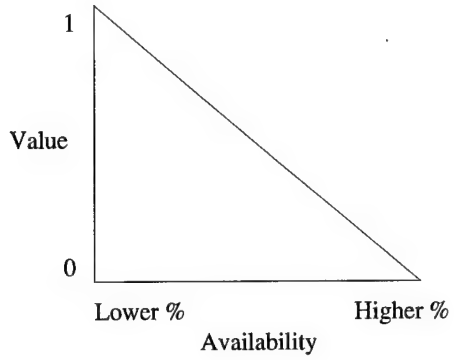


Figure 39 Availability Value Function

*Physical Defense* is used to measure the number of techniques used to protect the physical components of the system. These techniques can be any number of defense tactics: no defense, a locked door, a barb wire fence, a security gate, a guard, camouflage, ground, or others. The more defense tactics used the less vulnerable the system becomes. The value function is shown in Figure 40.

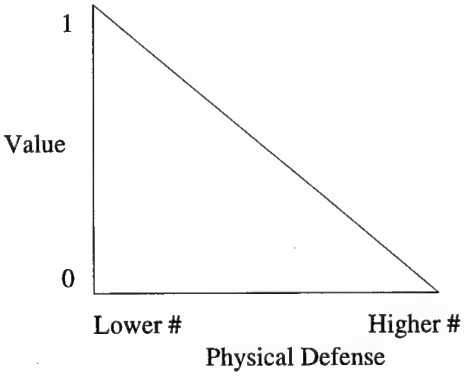


Figure 40 Physical Defense Value Function

## Appendix B Making Breakfast Network

This appendix shows a making breakfast network. The network is comprised of nodes and links defined as components necessary to make oatmeal and toast for breakfast. The analysis of this network is limited to proving the methodology of the value focused thinking techniques.

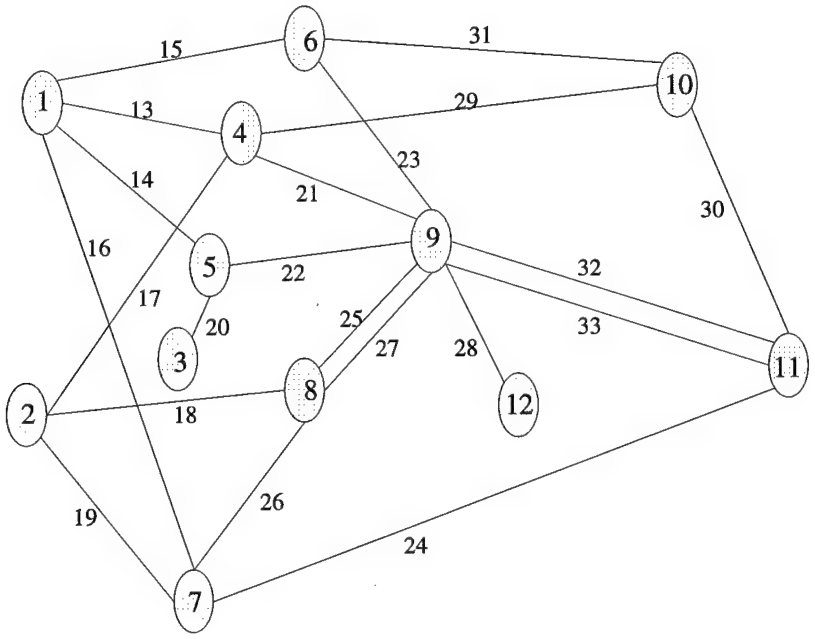


Figure 41 Making Breakfast Network

The network components are and characteristics are shown in Table 10. These characteristics are used in the Network Vulnerability Assessment Tool value model to determine each components score. For simplicity, a linear utility function is used for each measure. The vulnerabilities are ranked from highest to lowest score giving a prioritized list of the networks vulnerabilities.

Table 10 Making Breakfast Network Characteristics

| Nodes | Criticality | Impact Time | Ease of Transport | Repairable | Redundancy | Hardening | Physical Defense | Availability |
|---|---|---|---|---|---|---|---|---|
| 1 Power Supply | 7 | 0 | 4 | 72 | 0 | 0 | 2 | 0.95 |
| 2 Water Supply | 9 | 0.03 | 4 | 72 | 0 | 0 | 2 | 0.98 |
| 3 Gas Supply | 9 | 0 | 4 | 72 | 0 | 0 | 3 | 0.98 |
| 4 Refrigerator | 2 | 2 | 3 | 4 | 0 | 0 | 1 | 0.99 |
| 5 Stove | 9 | 0 | 3 | 4 | 0 | 0 | 1 | 0.99 |
| 6 Toaster | 3 | 0 | 2 | 1 | 1 | 0 | 1 | 0.98 |
| 7 Dishwasher | 1 | 0 | 3 | 4 | 1 | 0 | 1 | 0.99 |
| 8 Sink | 9 | 0 | 4 | 6 | 3 | 0 | 1 | 1 |
| 9 Cooked Food | 10 | 0 | 4 | 1 | 0 | 0 | 1 | 1 |
| 10 Store | 9 | 0 | 4 | 24 | 3 | 0 | 3 | 0.95 |
| 11 Cupboard | 1 | 0 | 4 | 12 | 1 | 0 | 1 | 1 |
| 12 Trash Can | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| Links | | | | | | | | |
| 13 Power | 2 | 0 | 4 | 24 | 0 | 0 | 2 | 0.95 |
| 14 Power | 7 | 0 | 4 | 24 | 0 | 0 | 2 | 0.95 |
| 15 Power | 3 | 0 | 4 | 24 | 3 | 0 | 2 | 0.95 |
| 16 Power | 1 | 0 | 4 | 24 | 0 | 0 | 2 | 0.95 |
| 17 Water | 2 | 0 | 4 | 24 | 0 | 0 | 2 | 0.98 |
| 18 Water | 9 | 0 | 4 | 24 | 0 | 0 | 2 | 0.98 |
| 19 Water | 1 | 0 | 4 | 24 | 0 | 0 | 2 | 0.98 |
| 20 Gas | 9 | 0 | 4 | 24 | 0 | 0 | 2 | 0.98 |
| 21 Butter | 2 | 0 | 1 | 0.02 | 0 | 0 | 1 | 0.99 |
| 22 Heat | 10 | 0 | 4 | 6 | 3 | 0 | 1 | 0.98 |
| 23 Toast | 3 | 0 | 1 | 0.02 | 0 | 0 | 1 | 0.99 |
| 24 Clean Dishes | 0 | 0 | 1 | 0.02 | 1 | 0 | 1 | 0.99 |
| 25 Water | 10 | 0 | 1 | 0.02 | 3 | 0 | 1 | 0.98 |
| 26 Dirty Dishes | 0 | 0 | 1 | 0.02 | 1 | 0 | 1 | 0.99 |
| 27 Dirty Dishes | 0 | 0 | 1 | 0.02 | 1 | 0 | 1 | 0.99 |
| 28 Trash | 0 | 0 | 1 | 0.02 | 1 | 0 | 1 | 0.99 |
| 29 Butter | 2 | 0 | 1 | 0.02 | 3 | 0 | 0 | 0.95 |
| 30 Oatmeal | 10 | 0 | 1 | 0.02 | 0 | 0 | 0 | 0.95 |
| 31 Bread | 10 | 0 | 1 | 0.02 | 11 | 0 | 0 | 0.95 |
| 32 Clean Dishes | 6 | 0 | 1 | 0.02 | 3 | 0 | 1 | 0.99 |
| 33 Oatmeal | 1 | 0 | 1 | 0.02 | 0 | 0 | 1 | 0.99 |

The vulnerability scores are shown in Table 11. According to the prioritized list shown in Table 12, the water supply, power supply, gas supply, and oatmeal are the most vulnerable components of the network. The least vulnerable components are the trashcan and refrigerator.

# Table 11 Making Breakfast Vulnerability Scores

| Weights | Functionality | | 0.5 | Flexibility | | 0.2 | Survivability 0.3 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | Media Protection | | 0.4 | Physical Endurance | | 0.6 | |
| | 0.6 | 0.4 | Weighted | 0.3 | 0.7 | Weighted | 0.6 | 0.4 | Weighted | 0.7 | 0.3 | Weighted | Total |
| Nodes | Crit. Score | Imp Time Scor | Score | EOT Score | Rep. Score | Score | Red. Score | Hard. Score | Score | Phy Def Score | Avail. Score | Score | Score |
| 1 Power Supply | 0.7 | 1 | 0.82 | 1 | 1 | 1 | 1 | 1 | 1 | 0.333 | 1 | 0.533 | 0.826 |
| 2 Water Supply | 0.9 | 0.985 | 0.934 | 1 | 1 | 1 | 1 | 1 | 1 | 0.333 | 0.4 | 0.353 | 0.851 |
| 3 Gas Supply | 0.9 | 1 | 0.94 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0.4 | 0.12 | 0.812 |
| 4 Refrigerator | 0.2 | 0 | 0.12 | 0.667 | 0.055 | 0.239 | 1 | 1 | 1 | 0.667 | 0.2 | 0.527 | 0.323 |
| 5 Stove | 0.9 | 1 | 0.94 | 0.667 | 0.055 | 0.239 | 1 | 1 | 1 | 0.667 | 0.2 | 0.527 | 0.733 |
| 6 Toaster | 0.3 | 1 | 0.58 | 0.333 | 0.014 | 0.110 | 0.909 | 1 | 0.945 | 0.667 | 0.4 | 0.587 | 0.531 |
| 7 Dishwasher | 0.1 | 1 | 0.46 | 0.667 | 0.055 | 0.239 | 0.909 | 1 | 0.945 | 0.667 | 0.2 | 0.527 | 0.486 |
| 8 Sink | 0.9 | 1 | 0.94 | 1 | 0.083 | 0.358 | 0.727 | 1 | 0.836 | 0.667 | 0 | 0.467 | 0.726 |
| 9 Cooked Food | 1 | 1 | 1 | 1 | 0.014 | 0.310 | 1 | 1 | 1 | 0.667 | 0 | 0.467 | 0.766 |
| 10 Store | 0.9 | 1 | 0.94 | 1 | 0.333 | 0.533 | 0.727 | 1 | 0.836 | 0 | 1 | 0.3 | 0.731 |
| 11 Cupboard | 0.1 | 1 | 0.46 | 1 | 0.166 | 0.417 | 0.909 | 1 | 0.945 | 0.667 | 0 | 0.467 | 0.511 |
| 12 Trash Can | 0 | 1 | 0.4 | 0 | 0.014 | 0.010 | 0.909 | 1 | 0.945 | 0.667 | 0 | 0.467 | 0.399 |
| Links | | | | | | | | | | | | | |
| 13 Power | 0.2 | 1 | 0.52 | 1 | 0.333 | 0.533 | 1 | 1 | 1 | 0.333 | 1 | 0.533 | 0.583 |
| 14 Power | 0.7 | 1 | 0.82 | 1 | 0.333 | 0.533 | 1 | 1 | 1 | 0.333 | 1 | 0.533 | 0.733 |
| 15 Power | 0.3 | 1 | 0.58 | 1 | 0.333 | 0.533 | 0.727 | 1 | 0.836 | 0.333 | 1 | 0.533 | 0.593 |
| 16 Power | 0.1 | 1 | 0.46 | 1 | 0.333 | 0.533 | 1 | 1 | 1 | 0.333 | 1 | 0.533 | 0.553 |
| 17 Water | 0.2 | 1 | 0.52 | 1 | 0.333 | 0.533 | 1 | 1 | 1 | 0.333 | 0.4 | 0.353 | 0.550 |
| 18 Water | 0.9 | 1 | 0.94 | 1 | 0.333 | 0.533 | 1 | 1 | 1 | 0.333 | 0.4 | 0.353 | 0.760 |
| 19 Water | 0.1 | 1 | 0.46 | 1 | 0.333 | 0.533 | 1 | 1 | 1 | 0.333 | 0.4 | 0.353 | 0.520 |
| 20 Gas | 0.9 | 1 | 0.94 | 1 | 0.333 | 0.533 | 1 | 1 | 1 | 0.333 | 0.4 | 0.353 | 0.760 |
| 21 Butter | 0.2 | 1 | 0.52 | 0 | 0 | 0 | 1 | 1 | 1 | 0.667 | 0.2 | 0.527 | 0.475 |
| 22 Heat | 1 | 1 | 1 | 1 | 0.083 | 0.358 | 0.727 | 1 | 0.836 | 0.667 | 0.4 | 0.587 | 0.778 |
| 23 Toast | 0.3 | 1 | 0.58 | 0 | 0 | 0 | 1 | 1 | 1 | 0.667 | 0.2 | 0.527 | 0.505 |
| 24 Clean Dishes | 0 | 1 | 0.4 | 0 | 0 | 0 | 0.909 | 1 | 0.945 | 0.667 | 0.2 | 0.527 | 0.408 |
| 25 Water | 1 | 1 | 1 | 0 | 0 | 0 | 0.727 | 1 | 0.836 | 0.667 | 0.4 | 0.587 | 0.706 |
| 26 Dirty Dishes | 0 | 1 | 0.4 | 0 | 0 | 0 | 0.909 | 1 | 0.945 | 0.667 | 0.2 | 0.527 | 0.408 |
| 27 Dirty Dishes | 0 | 1 | 0.4 | 0 | 0 | 0 | 0.909 | 1 | 0.945 | 0.667 | 0.2 | 0.527 | 0.408 |
| 28 Trash | 0 | 1 | 0.4 | 0 | 0 | 0 | 0.909 | 1 | 0.945 | 0.667 | 0.2 | 0.527 | 0.408 |
| 29 Butter | 0.2 | 1 | 0.52 | 0 | 0 | 0 | 0.727 | 1 | 0.836 | 1 | 1 | 1 | 0.540 |
| 30 Oatmeal | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0.8 |
| 31 Bread | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0.4 | 1 | 1 | 1 | 0.728 |
| 32 Clean Dishes | 0.6 | 1 | 0.76 | 0 | 0 | 0 | 0.727 | 1 | 0.836 | 0.667 | 0.2 | 0.527 | 0.575 |
| 33 Oatmeal | 0.1 | 1 | 0.46 | 0 | 0 | 0 | 1 | 1 | 1 | 0.667 | 0.2 | 0.527 | 0.445 |

Table 12 Prioritized Vulnerability List

| Nodes | Total Score |
|---|---|
| 2 Water Supply | 0.851 |
| 1 Power Supply | 0.826 |
| 3 Gas Supply | 0.812 |
| 30 Oatmeal | 0.8 |
| 22 Heat | 0.778 |
| 9 Cooked Food | 0.766 |
| 18 Water | 0.760 |
| 20 Gas | 0.760 |
| 14 Power | 0.733 |
| 5 Stove | 0.733 |
| 10 Store | 0.731 |
| 31 Bread | 0.728 |
| 8 Sink | 0.726 |
| 25 Water | 0.706 |
| 15 Power | 0.593 |
| 13 Power | 0.583 |
| 32 Clean Dishes | 0.575 |
| 16 Power | 0.553 |
| 17 Water | 0.550 |
| 29 Butter | 0.540 |
| 6 Toaster | 0.531 |
| 19 Water | 0.520 |
| 11 Cupboard | 0.511 |
| 23 Toast | 0.505 |
| 7 Dishwasher | 0.486 |
| 21 Butter | 0.475 |
| 33 Oatmeal | 0.445 |
| 24 Clean Dishes | 0.408 |
| 26 Dirty Dishes | 0.408 |
| 27 Dirty Dishes | 0.408 |
| 28 Trash | 0.408 |
| 12 Trash Can | 0.399 |
| 4 Refrigerator | 0.323 |

# Appendix C Notional Network Data

This appendix contains the complete data set of the notional network results.

Notional Network Node Attributes

| Component | Description | Criticality | Impact Time | Ease of Transport | Repairable | Redundancy | Hardening | Availability | Physical Defense |
|---|---|---|---|---|---|---|---|---|---|
| Node 2 | Soccorro Optical Sensor | 9 | 24 | 4 | 72 | 0 | 0 | 0.95 | 4 |
| Node 3 | Maui Optical Sensor | 9 | 24 | 4 | 72 | 0 | 0 | 0.93 | 1 |
| Node 4 | Diego Garcia Optical Sensor | 9 | 24 | 4 | 72 | 0 | 0 | 0.94 | 5 |
| Node 5 | Primary router | 9 | 24 | 3 | 24 | 0 | 0 | 0.98 | 2 |
| Node 6 | Primary router | 7 | 48 | 4 | 24 | 1 | 0 | 0.98 | 1 |
| Node 7 | Satellite | 6 | 48 | 2 | 12 | 1 | 1 | 0.95 | 3 |
| Node 8 | Primary router | 7 | 48 | 4 | 24 | 1 | 0 | 0.98 | 1 |
| Node 9 | Primary router | 7 | 48 | 3 | 24 | 1 | 0 | 0.98 | 1 |
| Node 10 | Tertiary router | 4 | 120 | 4 | 24 | 1 | 0 | 0.98 | 1 |
| Node 11 | OC3F-Edwards | 8 | 48 | 4 | 72 | 0 | 0 | 0.98 | 2 |
| Node 12 | Primary router | 8 | 48 | 4 | 24 | 0 | 0 | 0.98 | 1 |
| Node 13 | Peterson AFB | 8 | 24 | 4 | 48 | 0 | 0 | 0.98 | 3 |
| Node 14 | Shriever AFB | 6 | 24 | 4 | 48 | 0 | 0 | 0.98 | 4 |
| Node 15 | Secondary router | 6 | 48 | 3 | 24 | 1 | 0 | 0.98 | 1 |
| Node 16 | Colorado Springs router | 8 | 24 | 4 | 48 | 0 | 0 | 0.97 | 1 |
| Node 17 | Tertiary router | 4 | 120 | 4 | 24 | 1 | 0 | 0.98 | 1 |
| Node 18 | NAVSPACE | 4 | 120 | 4 | 24 | 1 | 0 | 0.98 | 1 |
| Node 19 | Cheyenne Mountain Air Station | 10 | 0 | 4 | 96 | 0 | 1 | 0.99 | 5 |

Notional Network Link Attributes

| Component | Description | Criticality | Impact Time | Ease of Transport | Repairable | Redundancy | Hardening | Availability | Physical Defense |
|---|---|---|---|---|---|---|---|---|---|
| Link 2 to 5 | Primary data line | 9 | 24 | 4 | 12 | 2 | 0 | 0.99 | 1 |
| Link 3 to 6 | Primary microwave line | 8 | 48 | 4 | 48 | 0 | 0 | 0.96 | 1 |
| Link 3 to 7 | Satellite relay | 6 | 48 | 3 | 12 | 2 | 0 | 0.97 | 2 |
| Link 4 to 7 | Satellite relay | 6 | 48 | 3 | 12 | 2 | 0 | 0.97 | 2 |
| Link 4 to 8 | Primary microwave line | 8 | 48 | 4 | 48 | 1 | 0 | 0.96 | 1 |
| Link 5 to 10 | Tertiary data line | 4 | 120 | 4 | 48 | 0 | 0 | 0.99 | 1 |
| Link 5 to 11 | Primary data line | 8 | 48 | 4 | 12 | 0 | 0 | 0.99 | 1 |
| Link 5 to 13 | Secondary data line | 6 | 48 | 4 | 48 | 0 | 0 | 0.99 | 1 |
| Link 6 to 10 | Tertiary data line | 4 | 120 | 4 | 48 | 0 | 0 | 0.99 | 1 |
| Link 6 to 11 | Primary data line | 8 | 48 | 4 | 48 | 0 | 0 | 0.99 | 1 |
| Link 6 to 13 | Secondary data line | 6 | 48 | 4 | 48 | 0 | 0 | 0.98 | 1 |
| Link 7 to 11 | Satellite relay | 6 | 48 | 3 | 12 | 2 | 0 | 0.97 | 2 |
| Link 7 to 14 | Satellite relay | 6 | 48 | 3 | 12 | 2 | 0 | 0.97 | 2 |
| Link 7 to 18 | Satellite relay | 3 | 120 | 3 | 12 | 2 | 0 | 0.97 | 2 |
| Link 8 to 9 | Primary data line | 8 | 48 | 4 | 48 | 1 | 0 | 0.99 | 1 |
| Link 9 to 11 | Primary data line | 8 | 48 | 4 | 12 | 0 | 0 | 0.99 | 1 |
| Link 9 to 13 | Secondary data line | 6 | 48 | 4 | 24 | 0 | 0 | 0.99 | 1 |
| Link 10 to 18 | Tertiary data line | 4 | 120 | 4 | 48 | 0 | 0 | 0.99 | 1 |
| Link 11 to 12 | Primary data line | 8 | 48 | 4 | 24 | 1 | 0 | 0.99 | 1 |
| Link 11 to 15 | Secondary data line | 6 | 72 | 4 | 48 | 0 | 0 | 0.98 | 1 |
| Link 11 to 17 | Tertiary data line | 4 | 120 | 4 | 48 | 0 | 0 | 0.99 | 1 |
| Link 12 to 13 | Primary data line | 8 | 48 | 4 | 24 | 1 | 0 | 0.98 | 1 |
| Link 13 to 16 | Primary data line | 8 | 24 | 4 | 24 | 1 | 0 | 0.98 | 1 |
| Link 14 to 16 | Secondary data line | 6 | 24 | 4 | 48 | 1 | 0 | 0.99 | 1 |
| Link 15 to 19 | Secondary data line | 6 | 48 | 4 | 48 | 0 | 0 | 0.99 | 1 |
| Link 16 to 19 | Primary data line | 8 | 24 | 4 | 48 | 1 | 0 | 0.98 | 1 |
| Link 17 to 18 | Tertiary data line | 4 | 120 | 4 | 48 | 0 | 0 | 0.99 | 1 |

Notional Sub-Network Attributes

| Component | Criticality | Timeliness | Ease of Transport | Repairable | Redundancy | Hardening | Reliability | Physical Defense |
|---|---|---|---|---|---|---|---|---|
| **Node 2** | 9 | 24 | 4 | 72 | 0 | 0 | 0.95 | 1 |
| 2W1 | 2 | 120 | 4 | 72 | 0 | 0 | 0.98 | 1 |
| 2W2 | 2 | 120 | 4 | 48 | 0 | 0 | 0.98 | 0 |
| 2P1 | 8 | 48 | 4 | 24 | 1 | 0 | 0.97 | 1 |
| Link 2W1 to 2W2 | 2 | 120 | 4 | 24 | 0 | 0 | 0.98 | 1 |
| Link 2W2 to 2 | 2 | 120 | 4 | 24 | 0 | 0 | 0.98 | 1 |
| Link 2P1 to 2 | 8 | 48 | 4 | 24 | 1 | 0 | 0.98 | 1 |
| **Node 3** | 9 | 24 | 4 | 72 | 0 | 0 | 0.93 | 1 |
| 3P1 | 8 | 48 | 4 | 72 | 0 | 0 | 0.95 | 1 |
| 3P2 | 8 | 48 | 4 | 48 | 1 | 0 | 0.96 | 0 |
| Link 3P1 to 3P2 | 8 | 48 | 4 | 24 | 0 | 0 | 0.98 | 0 |
| Link 3P2 to 3 | 8 | 48 | 4 | 24 | 1 | 0 | 0.98 | 0 |
| **Node 4** | 9 | 24 | 4 | 72 | 0 | 0 | 0.94 | 5 |
| 4W1 | 2 | 120 | 4 | 48 | 0 | 0 | 0.98 | 1 |
| 4P1 | 9 | 48 | 4 | 48 | 1 | 0 | 0.94 | 2 |
| Link 4W1 to 4 | 2 | 120 | 4 | 24 | 0 | 0 | 0.98 | 1 |
| Link 4P1 to 4 | 9 | 48 | 4 | 24 | 1 | 0 | 0.95 | 1 |
| **Node 11** | 8 | 48 | 4 | 72 | 0 | 0 | 0.98 | 2 |
| 11P1 | 8 | 48 | 4 | 24 | 0 | 0 | 0.98 | 2 |
| Link 11P1 to 11 | 8 | 48 | 4 | 24 | 0 | 0 | 0.98 | 1 |
| **Node 18** | 4 | 120 | 4 | 24 | 1 | 0 | 0.98 | 1 |
| 18W1 | 2 | 120 | 4 | 48 | 0 | 0 | 0.97 | 0 |
| 18P1 | 4 | 120 | 4 | 24 | 1 | 0 | 0.97 | 1 |
| Link 18W1 to 18 | 2 | 120 | 4 | 48 | 0 | 0 | 0.98 | 1 |
| Link 18P1 to 18 | 4 | 120 | 4 | 24 | 1 | 0 | 0.98 | 0 |

Notional Sub-Network Attributes (cont.)

| Component | Criticality | Timeliness | Ease of Transport | Repairable | Redundancy | Hardening | Reliability | Physical Defense |
|---|---|---|---|---|---|---|---|---|
| **Node 19** | 10 | 0 | 4 | 96 | 0 | 1 | 0.99 | 5 |
| 19P1 | 10 | 6 | 4 | 12 | 1 | 1 | 0.98 | 4 |
| 19P2 | 7 | 48 | 4 | 48 | 0 | 0 | 0.96 | 1 |
| 19P3 | 7 | 48 | 4 | 48 | 0 | 0 | 0.97 | 0 |
| 19W1 | 6 | 12 | 4 | 72 | 0 | 0 | 0.97 | 1 |
| 19W2 | 6 | 6 | 4 | 48 | 1 | 1 | 0.97 | 0 |
| Link 19P1 to 19 | 10 | 6 | 4 | 12 | 1 | 1 | 0.98 | 4 |
| Link 19P2 to 19P3 | 7 | 48 | 4 | 24 | 0 | 0 | 0.98 | 0 |
| Link 19P3 to 19 | 7 | 48 | 4 | 24 | 0 | 0 | 0.98 | 1 |
| Link 19W1 to 19W2 | 6 | 12 | 4 | 24 | 0 | 0 | 0.98 | 1 |
| Link 19W2 to 19 | 6 | 6 | 4 | 24 | 1 | 1 | 0.98 | 1 |

Notional Network Top Twenty Vulnerability Set Composition and Values

| Vulnerability Set | Components | | | | | Value |
|---|---|---|---|---|---|---|
| 32750 | Node 2<br>Soccorro Optical Sensor | Node 3<br>Maui Optical Sensor | Node 4<br>Diego Garcia Optical Sensor | | | 0.87 |
| 32752 | Node 3<br>Maui Optical Sensor | Node 4<br>Diego Garcia Optical Sensor | Node 5<br>Primary Router | | | 0.84 |
| 32751 | Link 2 to 5<br>Primary Data Line | Node 3<br>Maui Optical Sensor | Node 4<br>Diego Garcia Optical Sensor | | | 0.84 |
| 25729 | Node 2<br>Soccorro Optical Sensor | Node 3<br>Maui Optical Sensor | Link 4 to 7<br>Satellite Relay | Node 11<br>OC3F-Edwards | Node 19<br>C. Springs Router | 0.76 |
| 25657 | Node 2<br>Soccorro Optical Sensor | Node 3<br>Maui Optical Sensor | Link 4 to 7<br>Satellite Relay | Node 11<br>OC3F-Edwards | Node 16<br>CMAS | 0.75 |
| 25693 | Node 2<br>Soccorro Optical Sensor | Node 3<br>Maui Optical Sensor | Link 4 to 7<br>Satellite Relay | Node 11<br>OC3F-Edwards | Link 16 to 19<br>Primary Data Line | 0.75 |
| 25639 | Node 2<br>Soccorro Optical Sensor | Node 3<br>Maui Optical Sensor | Link 4 to 7<br>Satellite Relay | Link 9 to 11<br>Primary Data Line | Node 16<br>C. Springs Router | 0.74 |
| 25594 | Node 2<br>Soccorro Optical Sensor | Node 3<br>Maui Optical Sensor | Link 4 to 7<br>Satellite Relay | Node 11<br>OC3F-Edwards | Link 13 to 16<br>Primary Data Line | 0.74 |
| 30803 | Node 3<br>Maui Optical Sensor | Node 4<br>Diego Garcia Optical Sensor | Link 5 to 10<br>Tertiary Data Line | Node 11<br>OC3F-Edwards | Node 16<br>C. Springs Router | 0.74 |
| 30564 | Node 3<br>Maui Optical Sensor | Node 4<br>Diego Garcia Optical Sensor | Link 10 to 18<br>Tertiary Data Line | Node 11<br>OC3F-Edwards | Node 16<br>C. Springs Router | 0.74 |
| 25658 | Link 2 to 5<br>Primary Data Line | Node 3<br>Maui Optical Sensor | Link 4 to 7<br>Satellite Relay | Node 11<br>OC3F-Edwards | Node 16<br>C. Springs Router | 0.73 |
| 25730 | Link 2 to 5<br>Primary Data Line | Node 3<br>Maui Optical Sensor | Link 4 to 7<br>Satellite Relay | Node 11<br>OC3F-Edwards | Node 19<br>CMAS | 0.73 |
| 30804 | Node 3<br>Maui Optical Sensor | Node 4<br>Diego Garcia Optical Sensor | Link 5 to 10<br>Tertiary Data Line | Node 11<br>OC3F-Edwards | Link 16 to 19<br>Primary Data Line | 0.73 |
| 30566 | Node 3<br>Maui Optical Sensor | Node 4<br>Diego Garcia Optical Sensor | Link 10 to 18<br>Tertiary Data Line | Node 11<br>OC3F-Edwards | Link 16 to 19<br>Primary Data Line | 0.73 |

Notional Network Top Twenty Vulnerability Set Composition and Values (Cont.)

| Vulnerability Set | Components | | | | | Value |
|---|---|---|---|---|---|---|
| 30805 | Node 3<br>Maui Optical Sensor | Node 4<br>Diego Garcia Optical Sensor | Link 5 to 10<br>Tertiary Data Line | Node 11<br>OC3F-Edwards | Node 19<br>CMAS | 0.73 |
| 30568 | Node 3<br>Maui Optical Sensor | Node 4<br>Diego Garcia Optical Sensor | Link 10 to 18<br>Tertiary Data Line | Node 11<br>OC3F-Edwards | Node 19<br>CMAS | 0.73 |
| 25732 | Node 2<br>Soccorro Optical Sensor | Node 3<br>Maui Optical Sensor | Node 7<br>Satellite | Node 11<br>OC3F-Edwards | Node 19<br>CMAS | 0.72 |
| 25659 | Node 3<br>Maui Optical Sensor | Link 4 to 7<br>Satellite Relay | Node 5<br>Primary Router | Node 11<br>OC3F-Edwards | Node 16<br>C. Springs Router | 0.72 |
| 25731 | Node 3<br>Maui Optical Sensor | Link 4 to 7<br>Satellite Relay | Node 5<br>Primary Router | Node 11<br>OC3F-Edwards | Node 19<br>CMAS | 0.72 |
| 25695 | Node 3<br>Maui Optical Sensor | Link 4 to 7<br>Satellite Relay | Node 5<br>Primary Router | Node 11<br>OC3F-Edwards | Link 16 to 19<br>Primary Data Line | 0.71 |

Major Objective Vulnerability Set Value Composition

| Vulnerability Set | Functionality | Flexibility | Survivability | | Total |
|---|---|---|---|---|---|
| 32750 | 0.5 | 0.2 | 0.174 | | 0.874 |
| 32752 | 0.5 | 0.14 | 0.1992 | | 0.8392 |
| 32751 | 0.5 | 0.15 | 0.1884 | | 0.8384 |
| 25729 | 0.405 | 0.146 | 0.2076 | | 0.7586 |
| 25657 | 0.44 | 0.17 | 0.1452 | | 0.7552 |
| 25693 | 0.405 | 0.146 | 0.198 | | 0.749 |
| 25639 | 0.405 | 0.116 | 0.2184 | | 0.7394 |
| 25594 | 0.405 | 0.134 | 0.198 | | 0.737 |
| 30803 | 0.335 | 0.176 | 0.2244 | | 0.7354 |
| 30564 | 0.335 | 0.176 | 0.2244 | | 0.7354 |
| 25658 | 0.405 | 0.116 | 0.2112 | | 0.7322 |
| 25730 | 0.44 | 0.14 | 0.1488 | | 0.7288 |
| 30804 | 0.335 | 0.176 | 0.216 | | 0.727 |
| 30566 | 0.335 | 0.176 | 0.216 | | 0.727 |
| 30805 | 0.375 | 0.2 | 0.1512 | | 0.7262 |
| 30568 | 0.375 | 0.2 | 0.1512 | | 0.7262 |
| 25732 | 0.44 | 0.158 | 0.1236 | | 0.7216 |
| 25659 | 0.405 | 0.11 | 0.2064 | | 0.7214 |
| 25731 | 0.44 | 0.134 | 0.144 | | 0.718 |
| 25695 | 0.405 | 0.11 | 0.192 | | 0.707 |

Evaluation Measure Vulnerability Set Composition

| Vulnerability Set | Criticality | Impact Time | Ease of Transport | Repairable | Redundancy | Hardening | Availability | Physical Defense | Total |
|---|---|---|---|---|---|---|---|---|---|
| 32750 | 0.3 | 0.2 | 0.06 | 0.14 | 0.072 | 0.048 | 0.054 | 0 | 0.874 |
| 32752 | 0.3 | 0.2 | 0.04 | 0.1 | 0.072 | 0.048 | 0.0432 | 0.036 | 0.8392 |
| 32751 | 0.3 | 0.2 | 0.06 | 0.09 | 0.0432 | 0.048 | 0.0432 | 0.054 | 0.8384 |
| 25729 | 0.23 | 0.175 | 0.048 | 0.098 | 0.0552 | 0.048 | 0.0324 | 0.072 | 0.7586 |
| 25657 | 0.255 | 0.185 | 0.048 | 0.122 | 0.0552 | 0.0288 | 0.0324 | 0.0288 | 0.7552 |
| 25693 | 0.23 | 0.175 | 0.048 | 0.098 | 0.0456 | 0.048 | 0.0324 | 0.072 | 0.749 |
| 25639 | 0.23 | 0.175 | 0.048 | 0.068 | 0.0552 | 0.048 | 0.0324 | 0.0828 | 0.7394 |
| 25594 | 0.23 | 0.175 | 0.048 | 0.086 | 0.0456 | 0.048 | 0.0324 | 0.072 | 0.737 |
| 30803 | 0.2 | 0.135 | 0.06 | 0.116 | 0.072 | 0.048 | 0.0324 | 0.072 | 0.7354 |
| 30564 | 0.2 | 0.135 | 0.06 | 0.116 | 0.072 | 0.048 | 0.0324 | 0.072 | 0.7354 |
| 25658 | 0.23 | 0.175 | 0.048 | 0.068 | 0.0372 | 0.048 | 0.0216 | 0.1044 | 0.7322 |
| 25730 | 0.255 | 0.185 | 0.048 | 0.092 | 0.0372 | 0.0288 | 0.0216 | 0.0612 | 0.7288 |
| 30804 | 0.2 | 0.135 | 0.06 | 0.116 | 0.0636 | 0.048 | 0.0324 | 0.072 | 0.727 |
| 30566 | 0.2 | 0.135 | 0.06 | 0.116 | 0.0636 | 0.048 | 0.0324 | 0.072 | 0.727 |
| 30805 | 0.23 | 0.145 | 0.06 | 0.14 | 0.072 | 0.0288 | 0.0216 | 0.0288 | 0.7262 |
| 30568 | 0.23 | 0.145 | 0.06 | 0.14 | 0.072 | 0.0288 | 0.0216 | 0.0288 | 0.7262 |
| 25732 | 0.255 | 0.185 | 0.036 | 0.122 | 0.0636 | 0.0096 | 0.0324 | 0.018 | 0.7216 |
| 25659 | 0.23 | 0.175 | 0.036 | 0.074 | 0.0432 | 0.048 | 0.0216 | 0.0936 | 0.7214 |
| 25731 | 0.255 | 0.185 | 0.036 | 0.098 | 0.0432 | 0.0288 | 0.0216 | 0.0504 | 0.718 |
| 25695 | 0.23 | 0.175 | 0.036 | 0.074 | 0.0288 | 0.048 | 0.0216 | 0.0936 | 0.707 |

Sensitivity Analysis

| Functionality | 0 | 0.1 | 0.2 | 0.3 | 0.4 | **0.5** | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Vulnerability Set 32750 | 0.75 | 0.77 | 0.80 | 0.82 | 0.85 | **0.87** | 0.90 | 0.92 | 0.95 | 0.97 | 1.00 |
| Vulnerability Set 32752 | 0.68 | 0.71 | 0.74 | 0.78 | 0.81 | **0.84** | 0.87 | 0.90 | 0.94 | 0.97 | 1.00 |
| Vulnerability Set 32751 | 0.68 | 0.71 | 0.74 | 0.77 | 0.81 | **0.84** | 0.87 | 0.90 | 0.94 | 0.97 | 1.00 |
| Vulnerability Set 25729 | 0.63 | 0.66 | 0.68 | 0.71 | 0.73 | **0.76** | 0.78 | 0.81 | 0.84 | 0.86 | 0.89 |
| Vulnerability Set 25657 | 0.71 | 0.72 | 0.73 | 0.74 | 0.75 | **0.75** | 0.76 | 0.77 | 0.78 | 0.79 | 0.80 |

| Flexibility | 0 | 0.1 | **0.2** | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Vulnerability Set 32750 | 0.57 | 0.86 | **0.87** | 0.89 | 0.91 | 0.92 | 0.94 | 0.95 | 0.98 | 0.98 | 1.00 |
| Vulnerability Set 32752 | 0.57 | 0.86 | **0.84** | 0.82 | 0.81 | 0.79 | 0.77 | 0.76 | 0.75 | 0.72 | 0.70 |
| Vulnerability Set 32751 | 0.52 | 0.85 | **0.84** | 0.83 | 0.83 | 0.81 | 0.80 | 0.79 | 0.78 | 0.76 | 0.75 |
| Vulnerability Set 25729 | 0.47 | 0.75 | **0.76** | 0.77 | 0.79 | 0.79 | 0.80 | 0.82 | 0.83 | 0.84 | 0.85 |
| Vulnerability Set 25657 | 0.43 | 0.76 | **0.75** | 0.75 | 0.76 | 0.75 | 0.74 | 0.74 | 0.75 | 0.73 | 0.73 |

| Redundancy | 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | **0.6** | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Vulnerability Set 32750 | 0.87 | 0.87 | 0.87 | 0.87 | 0.87 | 0.87 | **0.87** | 0.87 | 0.87 | 0.87 | 0.87 |
| Vulnerability Set 32752 | 0.84 | 0.84 | 0.84 | 0.84 | 0.84 | 0.84 | **0.84** | 0.84 | 0.84 | 0.84 | 0.84 |
| Vulnerability Set 32751 | 0.87 | 0.86 | 0.86 | 0.85 | 0.85 | 0.84 | **0.84** | 0.83 | 0.83 | 0.82 | 0.82 |
| Vulnerability Set 25729 | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 0.76 | **0.76** | 0.76 | 0.76 | 0.76 | 0.77 |
| Vulnerability Set 25657 | 0.77 | 0.77 | 0.77 | 0.76 | 0.76 | 0.76 | **0.75** | 0.75 | 0.75 | 0.75 | 0.74 |

| Availability | 0 | 0.1 | 0.2 | **0.3** | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Vulnerability Set 32750 | 0.82 | 0.84 | 0.86 | **0.87** | 0.89 | 0.91 | 0.93 | 0.95 | 0.96 | 0.98 | 0.82 |
| Vulnerability Set 32752 | 0.76 | 0.82 | 0.83 | **0.84** | 0.85 | 0.86 | 0.87 | 0.88 | 0.89 | 0.90 | 0.76 |
| Vulnerability Set 32751 | 0.74 | 0.83 | 0.83 | **0.84** | 0.85 | 0.85 | 0.86 | 0.87 | 0.87 | 0.88 | 0.74 |
| Vulnerability Set 25729 | 0.70 | 0.74 | 0.75 | **0.76** | 0.77 | 0.77 | 0.78 | 0.79 | 0.79 | 0.80 | 0.70 |
| Vulnerability Set 25657 | 0.65 | 0.75 | 0.75 | **0.75** | 0.76 | 0.76 | 0.76 | 0.76 | 0.76 | 0.76 | 0.65 |

| Survivability | 0 | 0.1 | 0.2 | **0.3** | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Vulnerability Set 32750 | 1.00 | 0.96 | 0.92 | **0.87** | 0.83 | 0.79 | 0.75 | 0.71 | 0.66 | 0.62 | 0.58 |
| Vulnerability Set 32752 | 0.91 | 0.89 | 0.86 | **0.84** | 0.81 | 0.79 | 0.77 | 0.74 | 0.71 | 0.69 | 0.66 |
| Vulnerability Set 32751 | 0.93 | 0.90 | 0.87 | **0.84** | 0.81 | 0.78 | 0.75 | 0.72 | 0.69 | 0.66 | 0.63 |
| Vulnerability Set 25729 | 0.88 | 0.84 | 0.80 | **0.76** | 0.72 | 0.68 | 0.64 | 0.60 | 0.56 | 0.52 | 0.48 |
| Vulnerability Set 25657 | 0.78 | 0.77 | 0.76 | **0.75** | 0.75 | 0.74 | 0.73 | 0.72 | 0.71 | 0.70 | 0.69 |

| Criticality | 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | **0.6** | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Vulnerability Set 32750 | 0.87 | 0.87 | 0.87 | 0.87 | 0.87 | 0.87 | **0.87** | 0.87 | 0.87 | 0.87 | 0.87 |
| Vulnerability Set 32752 | 0.84 | 0.84 | 0.84 | 0.84 | 0.84 | 0.84 | **0.84** | 0.84 | 0.84 | 0.84 | 0.84 |
| Vulnerability Set 32751 | 0.84 | 0.84 | 0.84 | 0.84 | 0.84 | 0.84 | **0.84** | 0.84 | 0.84 | 0.84 | 0.84 |
| Vulnerability Set 25729 | 0.78 | 0.78 | 0.77 | 0.77 | 0.77 | 0.76 | **0.76** | 0.75 | 0.75 | 0.75 | 0.74 |
| Vulnerability Set 25657 | 0.79 | 0.78 | 0.78 | 0.77 | 0.77 | 0.76 | **0.75** | 0.75 | 0.74 | 0.74 | 0.73 |

| Ease of Transport | 0 | 0.1 | 0.2 | **0.3** | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Vulnerability Set 32750 | 0.87 | 0.87 | 0.87 | **0.87** | 0.87 | 0.87 | 0.87 | 0.87 | 0.87 | 0.87 | 0.87 |
| Vulnerability Set 32752 | 0.84 | 0.84 | 0.84 | **0.84** | 0.84 | 0.84 | 0.84 | 0.84 | 0.83 | 0.83 | 0.83 |
| Vulnerability Set 32751 | 0.82 | 0.82 | 0.83 | **0.84** | 0.85 | 0.85 | 0.86 | 0.87 | 0.87 | 0.88 | 0.89 |
| Vulnerability Set 25729 | 0.76 | 0.76 | 0.76 | **0.76** | 0.76 | 0.76 | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 |
| Vulnerability Set 25657 | 0.75 | 0.75 | 0.75 | **0.75** | 0.76 | 0.76 | 0.76 | 0.76 | 0.76 | 0.77 | 0.77 |

| Media Protection | 0 | 0.1 | 0.2 | 0.3 | **0.4** | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Vulnerability Set 32750 | 0.79 | 0.81 | 0.83 | 0.85 | **0.87** | 0.90 | 0.92 | 0.94 | 0.96 | 0.98 | 1.00 |
| Vulnerability Set 32752 | 0.77 | 0.79 | 0.81 | 0.82 | **0.84** | 0.86 | 0.87 | 0.89 | 0.91 | 0.92 | 0.94 |
| Vulnerability Set 32751 | 0.81 | 0.82 | 0.83 | 0.83 | **0.84** | 0.85 | 0.85 | 0.86 | 0.87 | 0.87 | 0.88 |
| Vulnerability Set 25729 | 0.72 | 0.73 | 0.74 | 0.75 | **0.76** | 0.77 | 0.78 | 0.79 | 0.80 | 0.81 | 0.82 |
| Vulnerability Set 25657 | 0.72 | 0.73 | 0.74 | 0.75 | **0.75** | 0.76 | 0.77 | 0.78 | 0.79 | 0.80 | 0.80 |

Persistency Analysis

| Functionality | 0 | 0.1 | 0.2 | 0.3 | 0.4 | **0.5** | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Node 2 | 0 | 1 | 3 | 4 | 5 | **10** | 12 | 10 | 10 | 7 | 7 |
| Node 3 | 20 | 20 | 20 | 20 | 20 | **20** | 20 | 20 | 20 | 15 | 15 |
| Node 4 | 19 | 19 | 17 | 17 | 16 | **7** | 3 | 3 | 3 | 3 | 3 |
| Node 5 | 0 | 0 | 1 | 1 | 1 | **3** | 4 | 5 | 5 | 7 | 6 |
| Node 7 | 0 | 0 | 0 | 0 | 0 | **0** | 2 | 6 | 6 | 11 | 11 |
| Node 10 | 0 | 1 | 2 | 2 | 1 | **0** | 0 | 0 | 0 | 0 | 0 |
| Node 11 | 7 | 10 | 15 | 15 | 14 | **13** | 10 | 10 | 10 | 9 | 8 |
| Node 12 | 5 | 0 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 0 |
| Node 13 | 0 | 0 | 2 | 2 | 2 | **1** | 1 | 1 | 1 | 0 | 0 |
| Node 16 | 10 | 12 | 6 | 6 | 7 | **6** | 4 | 2 | 2 | 0 | 0 |
| Node 18 | 1 | 1 | 2 | 1 | 0 | **0** | 0 | 0 | 0 | 0 | 0 |
| Node 19 | 0 | 0 | 0 | 0 | 3 | **6** | 8 | 12 | 12 | 17 | 17 |
| Link 2 to 5 | 0 | 0 | 1 | 1 | 1 | **3** | 4 | 5 | 5 | 6 | 7 |
| Link 3 to 6 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 5 | 5 |
| Link 4 to 7 | 0 | 0 | 1 | 2 | 4 | **13** | 15 | 11 | 11 | 6 | 6 |
| Link 4 to 8 | 1 | 1 | 2 | 1 | 0 | **1** | 1 | 0 | 0 | 0 | 0 |
| Link 5 to 10 | 8 | 8 | 6 | 6 | 6 | **2** | 0 | 0 | 0 | 0 | 0 |
| Link 5 to 11 | 0 | 2 | 2 | 2 | 2 | **0** | 0 | 0 | 0 | 0 | 0 |
| Link 5 to 13 | 8 | 3 | 3 | 2 | 0 | **0** | 0 | 0 | 0 | 0 | 0 |
| Link 9 to 11 | 0 | 0 | 0 | 0 | 1 | **3** | 6 | 7 | 7 | 8 | 9 |
| Link 10 to 18 | 11 | 9 | 6 | 6 | 6 | **2** | 0 | 0 | 0 | 0 | 0 |
| Link 11 to 15 | 7 | 3 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 0 |
| Link 11 to 17 | 7 | 4 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 0 |
| Link 13 to 16 | 0 | 2 | 2 | 2 | 2 | **1** | 1 | 1 | 1 | 0 | 0 |
| Link 14 to 16 | 1 | 1 | 2 | 1 | 0 | **0** | 0 | 0 | 0 | 0 | 0 |
| Link 15 to 19 | 6 | 4 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 0 |
| Link 16 to 19 | 2 | 2 | 3 | 4 | 3 | **2** | 2 | 1 | 1 | 0 | 0 |
| Link 17 to 18 | 6 | 3 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 0 |

| Flexibility | 0 | 0.1 | **0.2** | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Node 2 | 5 | 9 | **10** | 7 | 5 | 2 | 2 | 2 | 2 | 1 | 1 |
| Node 3 | 19 | 20 | **20** | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| Node 4 | 0 | 3 | **7** | 12 | 16 | 19 | 19 | 19 | 19 | 20 | 20 |
| Node 5 | 9 | 7 | **3** | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| Node 7 | 8 | 0 | **0** | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Node 10 | 0 | 0 | **0** | 1 | 2 | 2 | 1 | 1 | 2 | 3 | 3 |
| Node 11 | 5 | 9 | **13** | 17 | 15 | 15 | 12 | 12 | 12 | 11 | 11 |
| Node 13 | 10 | 0 | **1** | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Node 16 | 0 | 6 | **6** | 3 | 4 | 3 | 2 | 2 | 2 | 2 | 2 |
| Node 17 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| Node 19 | 0 | 3 | **6** | 7 | 7 | 6 | 10 | 10 | 11 | 11 | 11 |
| Link 2 to 5 | 6 | 4 | **3** | 2 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| Link 3 to 6 | 1 | 0 | **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Link 4 to 7 | 12 | 17 | **13** | 7 | 3 | 1 | 1 | 1 | 1 | 0 | 0 |
| Link 4 to 8 | 0 | 1 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Link 5 to 10 | 0 | 0 | **2** | 4 | 5 | 7 | 8 | 8 | 8 | 8 | 8 |
| Link 5 to 11 | 0 | 0 | **0** | 0 | 2 | 2 | 2 | 2 | 2 | 0 | 0 |
| Link 5 to 13 | 0 | 0 | **0** | 0 | 0 | 2 | 2 | 2 | 2 | 2 | 2 |
| Link 9 to 11 | 15 | 7 | **3** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Link 9 to 13 | 3 | 0 | **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Link 10 to 18 | 0 | 0 | **2** | 4 | 6 | 7 | 8 | 8 | 8 | 8 | 8 |
| Link 11 to 17 | 0 | 0 | **0** | 0 | 0 | 0 | 2 | 2 | 3 | 3 | 3 |
| Link 13 to 16 | 7 | 2 | **1** | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 |
| Link 16 to 19 | 0 | 5 | **2** | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 |
| Link 17 to 18 | 0 | 0 | **0** | 0 | 0 | 0 | 2 | 2 | 2 | 3 | 3 |

| Survivability | 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Node 2 | 9 | 9 | 11 | **10** | 8 | 6 | 3 | 1 | 0 | 0 | 0 |
| Node 3 | 19 | 19 | 20 | **20** | 20 | 20 | 20 | 20 | 20 | 18 | 17 |
| Node 4 | 6 | 6 | 6 | **7** | 9 | 10 | 11 | 6 | 2 | 0 | 0 |
| Node 5 | 4 | 4 | 2 | **3** | 4 | 4 | 5 | 5 | 4 | 4 | 0 |
| Node 7 | 5 | 5 | 3 | **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Node 8 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| Node 9 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| Node 10 | 1 | 1 | 1 | **0** | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| Node 11 | 13 | 13 | 12 | **13** | 11 | 10 | 7 | 3 | 1 | 0 | 0 |
| Node 12 | 0 | 0 | 0 | **0** | 0 | 0 | 2 | 6 | 12 | 15 | 15 |
| Node 13 | 1 | 1 | 1 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Node 15 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| Node 16 | 1 | 1 | 2 | **6** | 9 | 11 | 12 | 9 | 4 | 2 | 5 |
| Node 18 | 0 | 0 | 0 | **0** | 0 | 0 | 3 | 10 | 16 | 14 | 7 |
| Node 19 | 14 | 14 | 11 | **6** | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Link 2 to 5 | 4 | 4 | 4 | **3** | 2 | 3 | 3 | 1 | 2 | 0 | 0 |
| Link 3 to 6 | 1 | 1 | 0 | **0** | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| Link 4 to 7 | 9 | 9 | 11 | **13** | 11 | 10 | 7 | 4 | 2 | 6 | 13 |
| Link 4 to 8 | 0 | 0 | 1 | **1** | 1 | 1 | 2 | 9 | 11 | 12 | 10 |
| Link 5 to 10 | 1 | 1 | 1 | **2** | 3 | 3 | 3 | 3 | 2 | 3 | 5 |
| Link 5 to 11 | 0 | 0 | 0 | **0** | 2 | 2 | 4 | 5 | 4 | 2 | 2 |
| Link 5 to 13 | 0 | 0 | 0 | **0** | 0 | 0 | 1 | 6 | 10 | 14 | 15 |
| Link 6 to 13 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Link 8 to 9 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Link 9 to 11 | 4 | 4 | 4 | **3** | 3 | 4 | 4 | 4 | 2 | 2 | 2 |
| Link 9 to 13 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 2 | 4 | 5 |
| Link 10 to 18 | 1 | 1 | 1 | **2** | 3 | 3 | 3 | 3 | 2 | 3 | 8 |
| Link 11 to 12 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| Link 11 to 15 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 5 | 8 | 8 | 10 |
| Link 11 to 17 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 0 | 2 | 6 |
| Link 12 to 13 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 2 | 2 | 2 | 0 |
| Link 13 to 16 | 0 | 0 | 1 | **1** | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Link 14 to 16 | 0 | 0 | 0 | **0** | 0 | 0 | 2 | 10 | 16 | 14 | 7 |
| Link 15 to 19 | 0 | 0 | 0 | **0** | 0 | 0 | 2 | 4 | 6 | 9 | 8 |
| Link 16 to 19 | 1 | 1 | 1 | **2** | 5 | 5 | 3 | 0 | 0 | 0 | 0 |
| Link 17 to 18 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 0 | 2 | 5 |

| Criticality | 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | **0.6** | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Node 2 | 12 | 12 | 12 | 12 | 12 | 12 | **10** | 9 | 9 | 9 | 7 |
| Node 3 | 20 | 20 | 20 | 20 | 20 | 20 | **20** | 20 | 20 | 20 | 20 |
| Node 4 | 3 | 3 | 3 | 3 | 5 | 5 | **7** | 9 | 9 | 10 | 12 |
| Node 5 | 4 | 4 | 4 | 4 | 3 | 3 | **3** | 3 | 3 | 2 | 2 |
| Node 7 | 2 | 2 | 2 | 2 | 2 | 2 | **0** | 0 | 0 | 0 | 0 |
| Node 10 | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 0 | 1 | 1 |
| Node 11 | 13 | 13 | 13 | 13 | 13 | 13 | **13** | 14 | 14 | 14 | 12 |
| Node 13 | 1 | 1 | 1 | 1 | 1 | 1 | **1** | 1 | 1 | 1 | 0 |
| Node 16 | 5 | 5 | 5 | 5 | 7 | 7 | **6** | 5 | 5 | 4 | 6 |
| Node 19 | 5 | 5 | 5 | 5 | 5 | 5 | **6** | 6 | 6 | 7 | 7 |
| Link 2 to 5 | 4 | 4 | 4 | 4 | 3 | 3 | **3** | 2 | 2 | 2 | 2 |
| Link 4 to 7 | 15 | 15 | 15 | 15 | 13 | 13 | **13** | 11 | 11 | 10 | 8 |
| Link 4 to 8 | 1 | 1 | 1 | 1 | 1 | 1 | **1** | 1 | 1 | 1 | 1 |
| Link 5 to 10 | 0 | 0 | 0 | 0 | 1 | 1 | **2** | 3 | 3 | 3 | 4 |
| Link 5 to 11 | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 2 |
| Link 9 to 11 | 3 | 3 | 3 | 3 | 3 | 3 | **3** | 2 | 2 | 2 | 2 |
| Link 10 to 18 | 0 | 0 | 0 | 0 | 1 | 1 | **2** | 3 | 3 | 3 | 4 |
| Link 13 to 16 | 1 | 1 | 1 | 1 | 1 | 1 | **1** | 1 | 1 | 1 | 0 |
| Link 16 to 19 | 4 | 4 | 4 | 4 | 2 | 2 | **2** | 3 | 3 | 3 | 3 |

| Ease of Transport | 0 | 0.1 | 0.2 | **0.3** | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Node 2 | 11 | 11 | 11 | **10** | 10 | 10 | 10 | 8 | 8 | 7 | 7 |
| Node 3 | 20 | 20 | 20 | **20** | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| Node 4 | 7 | 7 | 7 | **7** | 7 | 6 | 7 | 7 | 7 | 7 | 6 |
| Node 5 | 3 | 3 | 3 | **3** | 3 | 2 | 1 | 1 | 1 | 1 | 1 |
| Node 7 | 3 | 2 | 2 | **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Node 10 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Node 11 | 15 | 14 | 14 | **13** | 13 | 12 | 10 | 9 | 8 | 8 | 6 |
| Node 13 | 1 | 1 | 1 | **1** | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| Node 16 | 5 | 6 | 6 | **6** | 6 | 7 | 8 | 8 | 8 | 8 | 7 |
| Node 19 | 7 | 7 | 7 | **6** | 4 | 3 | 2 | 3 | 2 | 1 | 1 |
| Link 2 to 5 | 2 | 2 | 2 | **3** | 3 | 5 | 5 | 7 | 7 | 8 | 9 |
| Link 4 to 7 | 10 | 11 | 11 | **13** | 13 | 14 | 13 | 13 | 13 | 13 | 14 |
| Link 4 to 8 | 1 | 1 | 1 | **1** | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Link 5 to 10 | 2 | 2 | 2 | **2** | 2 | 1 | 2 | 2 | 2 | 2 | 1 |
| Link 5 to 11 | 0 | 0 | 0 | **0** | 0 | 0 | 2 | 2 | 2 | 2 | 3 |
| Link 9 to 11 | 1 | 2 | 2 | **3** | 3 | 4 | 4 | 6 | 7 | 7 | 8 |
| Link 10 to 18 | 2 | 2 | 2 | **2** | 2 | 2 | 2 | 2 | 2 | 2 | 1 |
| Link 13 to 16 | 1 | 1 | 1 | **1** | 1 | 1 | 2 | 2 | 3 | 4 | 4 |
| Link 16 to 19 | 2 | 1 | 1 | **2** | 4 | 4 | 3 | 4 | 4 | 4 | 4 |

| Redundancy | 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | **0.6** | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Node 2 | 10 | 10 | 10 | 10 | 10 | 10 | **10** | 11 | 10 | 10 | 8 |
| Node 3 | 20 | 20 | 20 | 20 | 20 | 20 | **20** | 20 | 20 | 20 | 20 |
| Node 4 | 3 | 3 | 3 | 5 | 5 | 6 | **7** | 7 | 8 | 8 | 10 |
| Node 5 | 3 | 3 | 3 | 3 | 4 | 3 | **3** | 3 | 3 | 3 | 3 |
| Node 7 | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 2 | 2 | 4 | 4 |
| Node 10 | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 1 | 1 | 1 |
| Node 11 | 10 | 10 | 11 | 13 | 13 | 13 | **13** | 14 | 15 | 14 | 13 |
| Node 13 | 1 | 1 | 1 | 1 | 1 | 1 | **1** | 1 | 1 | 1 | 0 |
| Node 16 | 5 | 5 | 5 | 6 | 6 | 6 | **6** | 6 | 6 | 5 | 4 |
| Node 19 | 1 | 1 | 2 | 2 | 3 | 4 | **6** | 7 | 8 | 10 | 12 |
| Link 2 to 5 | 7 | 7 | 7 | 5 | 4 | 4 | **3** | 2 | 2 | 2 | 2 |
| Link 4 to 7 | 17 | 17 | 17 | 15 | 15 | 14 | **13** | 11 | 10 | 8 | 6 |
| Link 4 to 8 | 2 | 2 | 2 | 1 | 1 | 1 | **1** | 1 | 0 | 0 | 0 |
| Link 5 to 10 | 0 | 0 | 0 | 1 | 1 | 1 | **2** | 2 | 2 | 2 | 3 |
| Link 5 to 11 | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 2 |
| Link 8 to 9 | 1 | 1 | 1 | 1 | 1 | 0 | **0** | 0 | 0 | 0 | 0 |
| Link 9 to 11 | 4 | 4 | 3 | 2 | 2 | 3 | **3** | 2 | 2 | 3 | 2 |
| Link 10 to 18 | 0 | 0 | 0 | 1 | 1 | 2 | **2** | 2 | 2 | 2 | 3 |
| Link 13 to 16 | 2 | 2 | 2 | 2 | 1 | 1 | **1** | 1 | 1 | 0 | 0 |
| Link 16 to 19 | 5 | 5 | 4 | 4 | 4 | 4 | **2** | 1 | 1 | 1 | 1 |

| Availability | 0 | 0.1 | 0.2 | **0.3** | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Node 2 | 9 | 8 | 10 | **10** | 11 | 11 | 11 | 12 | 12 | 12 | 9 |
| Node 3 | 19 | 20 | 20 | **20** | 20 | 20 | 20 | 19 | 19 | 19 | 19 |
| Node 4 | 8 | 5 | 5 | **7** | 8 | 8 | 8 | 9 | 9 | 9 | 8 |
| Node 5 | 4 | 5 | 4 | **3** | 2 | 2 | 2 | 2 | 2 | 2 | 4 |
| Node 7 | 4 | 0 | 0 | **0** | 1 | 2 | 2 | 2 | 3 | 3 | 4 |
| Node 10 | 1 | 0 | 0 | **0** | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| Node 11 | 12 | 13 | 13 | **13** | 13 | 14 | 14 | 12 | 13 | 13 | 12 |
| Node 13 | 1 | 1 | 1 | **1** | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Node 16 | 1 | 8 | 6 | **6** | 4 | 5 | 5 | 4 | 4 | 4 | 1 |
| Node 19 | 14 | 3 | 4 | **6** | 7 | 8 | 8 | 8 | 8 | 8 | 14 |
| Link 2 to 5 | 2 | 5 | 4 | **3** | 2 | 2 | 2 | 1 | 1 | 1 | 2 |
| Link 3 to 6 | 1 | 0 | 0 | **0** | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| Link 3 to 7 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| Link 4 to 7 | 8 | 15 | 15 | **13** | 11 | 10 | 10 | 9 | 8 | 8 | 8 |
| Link 4 to 8 | 0 | 0 | 1 | **1** | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| Link 5 to 10 | 2 | 1 | 1 | **2** | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Link 5 to 11 | 2 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| Link 9 to 11 | 3 | 4 | 3 | **3** | 3 | 2 | 2 | 3 | 2 | 2 | 3 |
| Link 10 to 18 | 2 | 1 | 1 | **2** | 3 | 2 | 2 | 2 | 2 | 2 | 2 |
| Link 13 to 16 | 0 | 1 | 1 | **1** | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| Link 16 to 19 | 1 | 4 | 4 | **2** | 3 | 1 | 1 | 1 | 1 | 1 | 1 |

| Media Protection | 0 | 0.1 | 0.2 | 0.3 | **0.4** | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Node 2 | 8 | 8 | 8 | 10 | **10** | 9 | 9 | 8 | 6 | 6 | 6 |
| Node 3 | 20 | 20 | 20 | 20 | **20** | 20 | 20 | 20 | 20 | 20 | 20 |
| Node 4 | 3 | 3 | 3 | 5 | **7** | 9 | 11 | 12 | 14 | 14 | 14 |
| Node 5 | 1 | 2 | 4 | 4 | **3** | 3 | 2 | 2 | 2 | 2 | 2 |
| Node 7 | 1 | 1 | 0 | 0 | **0** | 1 | 1 | 1 | 0 | 0 | 1 |
| Node 10 | 0 | 0 | 0 | 0 | **0** | 0 | 0 | 1 | 1 | 1 | 1 |
| Node 11 | 9 | 10 | 11 | 13 | **13** | 15 | 15 | 15 | 14 | 14 | 14 |
| Node 13 | 0 | 0 | 0 | 1 | **1** | 1 | 3 | 3 | 3 | 3 | 3 |
| Node 16 | 5 | 6 | 5 | 6 | **6** | 5 | 4 | 4 | 3 | 3 | 3 |
| Node 19 | 3 | 3 | 4 | 4 | **6** | 7 | 6 | 7 | 8 | 8 | 9 |
| Link 2 to 5 | 11 | 10 | 8 | 4 | **3** | 2 | 1 | 1 | 1 | 1 | 1 |
| Link 4 to 7 | 16 | 16 | 17 | 15 | **13** | 10 | 8 | 7 | 6 | 6 | 5 |
| Link 4 to 8 | 2 | 2 | 2 | 1 | **1** | 0 | 0 | 0 | 0 | 0 | 0 |
| Link 5 to 10 | 0 | 0 | 0 | 1 | **2** | 3 | 4 | 4 | 5 | 5 | 5 |
| Link 5 to 11 | 0 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 2 | 2 | 2 |
| Link 8 to 9 | 1 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 0 | 0 |
| Link 9 to 11 | 5 | 5 | 4 | 3 | **3** | 2 | 2 | 2 | 1 | 1 | 1 |
| Link 10 to 18 | 0 | 0 | 0 | 1 | **2** | 3 | 4 | 4 | 5 | 5 | 5 |
| Link 13 to 16 | 2 | 2 | 2 | 1 | **1** | 1 | 1 | 0 | 0 | 0 | 0 |
| Link 16 to 19 | 4 | 4 | 4 | 4 | **2** | 3 | 3 | 3 | 3 | 3 | 2 |

## Appendix D Visual Basic Code

This appendix contains all the visual basic modules used in the *Network Vulnerability Assessment Tool.*

# Cut-Set Module

```
'This module inputs the network topology and then uses the inversion method to find
'all the cut-sets and labels them as vulnerability sets. The algorithm was
'developed from the code written Fortran found in Bailey's thesis [1]. It was modified
'and adapted to eliminate the capacities and probabilities.

Const AC As Integer = 80, RW As Long = 100000, ND As Integer = 50
Dim NDTOT As Integer, N As Integer, T As Long, P As Integer
Dim Cut(RW, AC) As Long, Path(RW, AC) As Long
Dim Head(AC) As Integer, Tail(AC) As Integer
Dim I As Integer, J As Integer, R As Integer, C As Integer, Flag As Integer, CYC As Integer
Dim ARCCAP(ND, ND) As Integer, Mark(ND, ND) As Integer, MATCOL(ND, ND) As Integer
Dim INN As Integer, OUTT As Integer, XR(RW) As Long, XR1(RW) As Long, XB(RW) As Long,
XB1(RW) As Long
Dim Row(RW) As Long, Col(AC) As Integer, TT As Long
Dim K As Long, L As Integer, M As Integer, PP As Integer, SR As Integer, SK As Integer

Sub CutSet()

'Input Topology
With Sheets("Input")
   .Range("a2:a16384").ClearContents
   Sheets("Input").Activate

Input1:
      NDTOT = Application.InputBox("Enter the total number of nodes", "Network Components",
Type:=1)
      If NDTOT = False Then Exit Sub

MsgBox ("Note: Source ID number must be 1 and the Terminal node ID number must equal the total
number of nodes.")


'Identify Dummy Source and Sink Nodes

SR = 0
SK = 0

Input2:
      SR = Application.InputBox("Enter 1 if the Source node is a dummy node. Enter 0 otherwise.",
"Dummy Source", Type:=1)
      If SR = 1 Or SR = 0 Then
         GoTo Input3
        Else
         MsgBox ("That is not an option. Please re-enter.")
         GoTo Input2
      End If


Input3:
      SK = Application.InputBox("Enter 1 if the Sink node is a dummy node. Enter 0 otherwise.", "Dummy
Sink", Type:=1)
      If SK = 1 Or SK = 0 Then
```

```
        GoTo 10
       Else
        MsgBox ("That is not an option. Please re-enter.")
        GoTo Input3
     End If
10:

K = 2

For I = 1 To NDTOT
   ARCCAP(I, I) = 1
   If SR = 1 And I = 1 Or SK = 1 And I = NDTOT Then
      GoTo 20
     Else
      .Cells(K, 1) = "Node " & I
   K = K + 1
   End If
20:
Next I

'Arcs
If SR = 1 And SK = 1 Then
   K = NDTOT
End If

If SR = 1 And SK <> 1 Or SR <> 1 And SK = 1 Then
   K = NDTOT + 1
End If

If SR <> 1 And SK <> 1 Then
   K = NDTOT + 2
End If

With Sheets("Stats")
   .Cells(3, 2) = K
End With

Flag = 0
iAnswer = MsgBox("Click OK to enter Links. Click Cancel to exit routine.", vbOKCancel, "Network
Components")

If iAnswer = vbCancel Then Exit Sub

MsgBox ("When arc entry is completed enter 0 for the head and 0 for the tail to continue with routine.")

40:
Do While Flag <> 1

   If Flag = 0 Then

      I = Application.InputBox("Enter arc tail", "Link", Type:=1)
      J = Application.InputBox("Enter arc head", "Link", Type:=1)

      If J = 0 And I = 0 Then Exit Do
```

```
      If SR = 1 And I = 1 Or SK = 1 And J = NDTOT Then
         GoTo 50
        Else
         .Cells(K, 1) = "Link " & I & " to " & J
         K = K + 1
      End If
50:
      If I > NDTOT Or J > NDTOT Then
         MsgBox ("Node is out of range!")
         GoTo 40
      End If
         ARCCAP(I, J) = 1
   End If

Loop

End With


'Calculate Pathset/Cutset Matrices Columns

N = 1
For I = 1 To NDTOT
   For J = 1 To NDTOT
      If ARCCAP(I, J) > 0 Then
         MATCOL(I, J) = N
         Head(N) = J
         Tail(N) = I
         N = N + 1
      End If
   Next J
Next I
N = N - 1

'Calculate Acyclic paths from source to sink

P = 1
I = 1
J = 1
Flag = 0
CYC = 0

430:
If Flag = 0 Then

435:
   If J < NDTOT Then

      'Find next arc segment
      R = I
      C = J
      I = J
      J = 1
440:
```

```
      If ARCCAP(I, J) > 0 And Mark(I, J) = 0 And I <> J Or J > NDTOT Then
         GoTo 445
        Else
         J = J + 1
         GoTo 440
      End If

      'Record arc segment
445:
      If MATCOL(I, J) > 0 Then
         Path(P, MATCOL(I, J)) = ARCCAP(I, J)
      End If
      If ARCCAP(I, I) > 0 Then
         Path(P, MATCOL(I, I)) = ARCCAP(I, I)
      End If

      'Mark Row(I) to Guard Against Cycling

      If Row(I) = 1 Then
         J = NDTOT + 1
         CYC = 1
      End If
      Row(I) = 1
   GoTo 435
   End If

   'Mark Segment

   Mark(R, C) = 1

   'Clear Mark

   If I > 1 And CYC = 0 Then
      For L = 1 To NDTOT
         Mark(I, L) = 0
      Next L
   End If

   'Determine if Path exists

   If J = NDTOT Then
      P = P + 1
      Call ENDSNODE(I, J, NDTOT, Flag, CYC, Row)
     Else
      Call ENDSNODE(I, J, NDTOT, Flag, CYC, Row)
      For M = 1 To AC
         Path(P, M) = 0
      Next M
   End If
   I = 1
   J = 1
GoTo 430
End If
```

```
P = P - 1

'Eliminate Dummy Source and Dummy Sink Nodes

If SR = 1 Then
   For J = 1 To N
      If Tail(J) = 1 Then
         For I = 1 To P
            Path(I, J) = 0
         Next I
      End If
   Next J
End If

If SK = 1 Then
   For J = 1 To N
      If Head(J) = NDTOT Then
         For I = 1 To P
            Path(I, J) = 0
         Next I
      End If
   Next J
End If

'Reduce Path Matrix

'Zero out row(x) and col(x) arrays

For I = 1 To RW
   Row(I) = 0
Next I


For J = 1 To AC
   Col(J) = 0
Next J


'Identify zero columns in path matrix

For J = 1 To N
   I = 1
535:
   If Path(I, J) > 0 Or I > P Then
      GoTo 540
     Else
      I = I + 1
      GoTo 535
   End If
540:
   If I > P Then
      Col(J) = 1
   End If
Next J
```

```
'Pack Path Matrix

J = 1
555:
If J < N Then
   If Col(J) = 1 Then
      For K = J To N
         For I = 1 To P
            Path(I, K) = Path(I, K + 1)
         Next I
         Col(K) = Col(K + 1)
         Head(K) = Head(K + 1)
         Tail(K) = Tail(K + 1)
      Next K
      N = N - 1
      GoTo 555
     Else
      J = J + 1
      GoTo 555
     End If
End If

'Calculate min-cuts

'Initialize Matrix

T = 0
For J = 1 To N
   If Path(1, J) > 0 Then
      T = T + 1
      Cut(T, J) = Path(1, J)
   End If
Next J

'Loop control for multiplying path polynomial clear SW3 data arrays

MsgBox ("There are this many paths " & P)

With Sheets("Stats")
   .Cells(3, 6) = P
   .Cells(3, 4) = Date$
   .Cells(4, 4) = Time$
End With

For PP = 2 To P
   For I = 1 To RW
      Row(I) = 0
      XB(I) = 0
      XB1(I) = 0
      XR(I) = 0
      XR1(I) = 0
   Next I
```

```
For J = 1 To N
   Col(J) = 0
Next J

'Determine XR and XR1 members of A

For J = 1 To N
   If Path(PP, J) > 0 Then
      For I = 1 To T
         If Path(PP, J) = Cut(I, J) Then
            XR(I) = XR(I) + 1
            XR1(I) = J
         End If
      Next I
   End If
Next J

'Mark column of path to indicate XR1

For I = 1 To T
   If XR(I) = 1 Then
      Col(XR1(I)) = 3
   End If
Next I

'Determine XB

For I = 1 To T
   For J = 1 To N
      If Cut(I, J) > 0 Then
         XB(I) = XB(I) + 1
         XB1(I) = J
      End If
   Next J
Next I

For I = 1 To T
   If XB(I) = 1 And Path(PP, XB1(I)) > 0 Then
      Col(XB1(I)) = 1
     Else
      XB(I) = 0
   End If
Next I

'Multiply path to cutset

TT = T
For J = 1 To N

   'Avoid XB in A
   If Path(PP, J) > 0 And Col(J) <> 1 Then
      For I = 1 To TT

         'Avoid XB,XR,XR1 in A
```

```
        If XR(I) = 0 And XB(I) = 0 Then
          T = T + 1
          For L = 1 To N
            Cut(T, L) = Cut(I, L)
          Next L
          Cut(T, J) = Path(PP, J)

          'Identify residual XR1 in A
          If Col(J) = 3 Then
            XR1(T) = J
          End If
        End If
      Next I
    End If
Next J

'Check for XR1 Containment

For J = 1 To N
  For I = 1 To TT
    If XR1(I) = J Then
      For K = TT + 1 To T
        If XR1(K) = J Then
          OUTT = 0
          INN = 0
          L = 1
          Do While L < N
            If Cut(I, L) > Cut(K, L) Then
              OUTT = OUTT + 1
            Else
              If Cut(K, L) > Cut(I, L) Then
              INN = INN + 1
              End If
            End If
            If OUTT > 0 And INN > 0 Then
              L = N
            End If
            L = L + 1
          Loop
          If INN > 0 And OUTT = 0 Then
            Row(K) = 1
          End If
        End If
      Next K
    End If
  Next I
Next J

'Move XB, XR, XR1 Terms into A

For I = 1 To TT
  If XR(I) > 0 Or XB(I) > 0 Then
    T = T + 1
    For J = 1 To N
```

```
            Cut(T, J) = Cut(I, J)
        Next J
    End If
Next I

'Move A to top of cut matrix

K = 1
For I = TT + 1 To T
    For J = 1 To N
        Cut(K, J) = Cut(I, J)
    Next J
    Row(K) = Row(I)
    K = K + 1
Next I
T = K - 1

'Pack Cut

I = 1
Do While I < T
    If Row(I) > 0 Then
        For K = I + 1 To T
            For J = 1 To N
                Cut(K - 1, J) = Cut(K, J)
            Next J
            Row(K - 1) = Row(K)
        Next K
        I = I - 1
        T = T - 1
    End If
    I = I + 1
Loop

Next PP

Call ShowCutset

For I = 1 To RW
    Row(I) = 0
    For M = 1 To AC
        Path(I, M) = 0
        Cut(I, M) = 0
    Next M
Next I

For I = 1 To ND
    For J = 1 To ND
        Mark(I, J) = 0
        ARCCAP(I, J) = 0
        MATCOL(I, J) = 0
    Next J
Next I
```

```
OUTT = 0
INN = 0

With Sheets("Stats")
   .Cells(3, 5) = Date$
   .Cells(4, 5) = Time$
End With

End Sub

Sub ENDSNODE(I, J, NDTOT, Flag, CYC, Row)

Dim M As Integer

If I = 1 And CYC = 1 And J = NDTOT Then
   Flag = 1
End If

If I = 1 And CYC = 0 And J > NDTOT Then
   Flag = 1
End If

For M = 1 To NDTOT
   Row(M) = 0
Next M
CYC = 0

End Sub

Sub ShowCutset()

With Sheets("CutSets")
   .Range("a:iv").ClearContents

   Sheets("CutSets").Activate

MsgBox ("The total number of cuts is: " & T)

With Sheets("Stats")
   .Cells(3, 8) = T
End With

For I = 1 To T
  .Cells(I, 1) = "Vulnerability Set " & I
K = 2
   For J = 1 To N
      If Cut(I, J) > 0 Then
         If Tail(J) = Head(J) Then
            .Cells(I, K) = "Node " & Tail(J)
          Else
            .Cells(I, K) = "Link " & Tail(J) & " to " & Head(J)
         End If
         K = K + 1
      End If
```

```
    Next J
Next I

End With

End Sub
```

## Attribute Input Module

```
'This module is used to input the attributes for the evaluation measures
'of the network.

Sub AttributeInput()

Dim J As Integer, I As Integer

With Sheets("Input")
   .Range("B2:iv16384").ClearContents
   Sheets("Input").Activate

iAnswer1 = MsgBox("Would you like to keep the existing Values?", vbYesNo, "Values")
If iAnswer1 = vbYes Then GoTo 10
If iAnswer1 = vbNo Then
   MsgBox ("Please run the value model input prior to proceeding")
   Exit Sub
End If

10:

MsgBox ("Please Enter Attribute data for each network component on selected worksheet")
.Cells(2, 2).Activate

End With

End Sub
```

# Value Hierarchy Input Module

```
'This model is used to input a value hierarchy.  It allows
'a hierarchy with 4 levels to be input.  It refers to the
'value hierarchy by levels and values, not by values, sub-objectives
'and evaluation measures.


Sub ValueInput()

Dim I As Integer, A As Integer, L As Integer, M As Integer, N As Integer, J As Integer, K As Integer, B As
Integer, O As Integer, P As Integer, Q As Integer, R As Integer
Dim ValueA As String, Rho As Variant, level As Variant
Dim LevelA As Integer

levels = Application.InputBox("Enter the number of levels in the Value Hierarchy.", "Value Hierarchy",
Type:=1)
Sheets("Stats").Cells(3, 10).Value = levels

Sheets("Values").Range("A5:iv16834").ClearContents
Sheets("Sub-Net Values").Range("A5:iv16834").ClearContents
Sheets("Sensitivity Values").Range("A5:iv16834").ClearContents
Sheets("Input").Range("B1:Z1").ClearContents
Sheets("Sub-Net Input").Range("B1:Z1").ClearContents
Sheets("Scores").Range("B1:Z1").ClearContents

I = 1
A = 1
S = 1
Z = 2
LevelA = 2


J = InputBox("Enter number of values in level " & I)
For K = 1 To J
   Value = Application.InputBox("Enter value " & K & " in level " & I, "Value Hierarchy", Type:=2)
   Weight = Application.InputBox("Enter weight " & K & " in level " & I, "Value Hierarchy", Type:=1)
   Sheets("Values").Cells(5, S).Value = Value
   Sheets("Values").Cells(5, S + 1).Value = Weight
   Sheets("Sub-Net Values").Cells(5, S).Value = Value
   Sheets("Sub-Net Values").Cells(5, S + 1).Value = Weight
   Sheets("Sensitivity Values").Cells(5, S).Value = Value
   Sheets("Sensitivity Values").Cells(5, S + 1).Value = Weight
   L = InputBox("Enter number of values in level " & LevelA & " under value " & K & " in level " & I)
   If L = 0 Then
      Sheets("Input").Cells(1, Z).Value = Value
      Sheets("Sub-Net Input").Cells(1, Z).Value = Value
      Sheets("Scores").Cells(1, Z).Value = Value
      Z = Z + 1
      Sheets("Values").Cells(6 + levels, S) = "Low:"
      Sheets("Values").Cells(7 + levels, S) = "High:"
      Sheets("Values").Cells(8 + levels, S) = "Mono:"
      Sheets("Values").Cells(9 + levels, S) = "Rho:"
      Sheets("Values").Cells(10 + levels, S) = "Weight:"
```

```
        Sheets("Sub-Net Values").Cells(6 + levels, S) = "Low:"
        Sheets("Sub-Net Values").Cells(7 + levels, S) = "High:"
        Sheets("Sub-Net Values").Cells(8 + levels, S) = "Mono:"
        Sheets("Sub-Net Values").Cells(9 + levels, S) = "Rho:"
        Sheets("Sub-Net Values").Cells(10 + levels, S) = "Weight:"
        Sheets("Sensitivity Values").Cells(6 + levels, S) = "Low:"
        Sheets("Sensitivity Values").Cells(7 + levels, S) = "High:"
        Sheets("Sensitivity Values").Cells(8 + levels, S) = "Mono:"
        Sheets("Sensitivity Values").Cells(9 + levels, S) = "Rho:"
        Sheets("Sensitivity Values").Cells(10 + levels, S) = "Weight:"
        Mon = Application.InputBox("Enter the monotonicity: Increasing or Decreasing", Type:=2)
        Rho = InputBox("Enter Rho")
        Sheets("Values").Cells(8 + levels, S + 1) = Mon
        Sheets("Values").Cells(9 + levels, S + 1) = Rho
        Sheets("Values").Cells(10 + levels, S + 1) = Weight
        Sheets("Sub-Net Values").Cells(8 + levels, S + 1) = Mon
        Sheets("Sub-Net Values").Cells(9 + levels, S + 1) = Rho
        Sheets("Sub-Net Values").Cells(10 + levels, S + 1) = Weight
        Sheets("Sensitivity Values").Cells(8 + levels, S + 1) = Mon
        Sheets("Sensitivity Values").Cells(9 + levels, S + 1) = Rho
        Sheets("Sensitivity Values").Cells(10 + levels, S + 1) = Weight
    Else
      For M = 1 To L
        Value = Application.InputBox("Enter value " & M & " in level " & LevelA & " under value " & K
& " in level " & I, "Value Hierarchy", Type:=2)
        Weight = Application.InputBox("Enter weight " & M & " in level " & LevelA & " under value " &
K & " in level " & I, "Value Hierarchy", Type:=1)
        Sheets("Values").Cells(6, S).Value = Value
        Sheets("Values").Cells(6, S + 1).Value = Weight
        Sheets("Sub-Net Values").Cells(6, S).Value = Value
        Sheets("Sub-Net Values").Cells(6, S + 1).Value = Weight
        Sheets("Sensitivity Values").Cells(6, S).Value = Value
        Sheets("Sensitivity Values").Cells(6, S + 1).Value = Weight
        N = InputBox("Enter number of values in level " & LevelA + 1 & " under value " & M & " in level
" & LevelA & " under value " & K & " in level " & I)
        If N = 0 Then
          Sheets("Input").Cells(1, Z).Value = Value
          Sheets("Sub-Net Input").Cells(1, Z).Value = Value
          Sheets("Scores").Cells(1, Z).Value = Value
          Z = Z + 1
          Sheets("Values").Cells(6 + levels, S) = "Low:"
          Sheets("Values").Cells(7 + levels, S) = "High:"
          Sheets("Values").Cells(8 + levels, S) = "Mono:"
          Sheets("Values").Cells(9 + levels, S) = "Rho:"
          Sheets("Values").Cells(10 + levels, S) = "Weight:"
          Sheets("Sub-Net Values").Cells(6 + levels, S) = "Low:"
          Sheets("Sub-Net Values").Cells(7 + levels, S) = "High:"
          Sheets("Sub-Net Values").Cells(8 + levels, S) = "Mono:"
          Sheets("Sub-Net Values").Cells(9 + levels, S) = "Rho:"
          Sheets("Sub-Net Values").Cells(10 + levels, S) = "Weight:"
          Sheets("Sensitivity Values").Cells(6 + levels, S) = "Low:"
          Sheets("Sensitivity Values").Cells(7 + levels, S) = "High:"
          Sheets("Sensitivity Values").Cells(8 + levels, S) = "Mono:"
          Sheets("Sensitivity Values").Cells(9 + levels, S) = "Rho:"
```

```
            Sheets("Sensitivity Values").Cells(10 + levels, S) = "Weight:"
            Mon = Application.InputBox("Enter the monotonicity: Increasing or Decreasing", Type:=2)
            Rho = InputBox("Enter Rho")
            Sheets("Values").Cells(8 + levels, S + 1) = Mon
            Sheets("Values").Cells(9 + levels, S + 1) = Rho
            Sheets("Values").Cells(10 + levels, S + 1) = Weight
            Sheets("Sub-Net Values").Cells(8 + levels, S + 1) = Mon
            Sheets("Sub-Net Values").Cells(9 + levels, S + 1) = Rho
            Sheets("Sub-Net Values").Cells(10 + levels, S + 1) = Weight
            Sheets("Sensitivity Values").Cells(8 + levels, S + 1) = Mon
            Sheets("Sensitivity Values").Cells(9 + levels, S + 1) = Rho
            Sheets("Sensitivity Values").Cells(10 + levels, S + 1) = Weight
          Else
            For O = 1 To N
              Value = Application.InputBox("Enter value " & O & " in level " & LevelA + 1 & " under
value " & M & " in level " & LevelA & " under value " & K & " in level " & I, "Value Hierarchy",
Type:=2)
              Weight = Application.InputBox("Enter weight " & O & " in level " & LevelA + 1 & " under
value " & M & " in level " & LevelA & " under value " & K & " in level " & I, "Value Hierarchy",
Type:=1)
              Sheets("Values").Cells(7, S).Value = Value
              Sheets("Values").Cells(7, S + 1).Value = Weight
              Sheets("Sub-Net Values").Cells(7, S).Value = Value
              Sheets("Sub-Net Values").Cells(7, S + 1).Value = Weight
              Sheets("Sensitivity Values").Cells(7, S).Value = Value
              Sheets("Sensitivity Values").Cells(7, S + 1).Value = Weight
              P = InputBox("Enter number of values in level " & LevelA + 2 & " under value " & O & " in
level " & LevelA + 1 & " under value " & M & " in level " & LevelA & " under value " & K & " in level "
& I)
              If P = 0 Then
                Sheets("Input").Cells(1, Z).Value = Value
                Sheets("Sub-Net Input").Cells(1, Z).Value = Value
                Sheets("Scores").Cells(1, Z).Value = Value
                Z = Z + 1
                Sheets("Values").Cells(6 + levels, S) = "Low:"
                Sheets("Values").Cells(7 + levels, S) = "High:"
                Sheets("Values").Cells(8 + levels, S) = "Mono:"
                Sheets("Values").Cells(9 + levels, S) = "Rho:"
                Sheets("Values").Cells(10 + levels, S) = "Weight:"
                Sheets("Sub-Net Values").Cells(6 + levels, S) = "Low:"
                Sheets("Sub-Net Values").Cells(7 + levels, S) = "High:"
                Sheets("Sub-Net Values").Cells(8 + levels, S) = "Mono:"
                Sheets("Sub-Net Values").Cells(9 + levels, S) = "Rho:"
                Sheets("Sub-Net Values").Cells(10 + levels, S) = "Weight:"
                Sheets("Sensitivity Values").Cells(6 + levels, S) = "Low:"
                Sheets("Sensitivity Values").Cells(7 + levels, S) = "High:"
                Sheets("Sensitivity Values").Cells(8 + levels, S) = "Mono:"
                Sheets("Sensitivity Values").Cells(9 + levels, S) = "Rho:"
                Sheets("Sensitivity Values").Cells(10 + levels, S) = "Weight:"
                Mon = Application.InputBox("Enter the monotonicity: Increasing or Decreasing", Type:=2)
                Rho = InputBox("Enter Rho")
                Sheets("Values").Cells(8 + levels, S + 1) = Mon
                Sheets("Values").Cells(9 + levels, S + 1) = Rho
                Sheets("Values").Cells(10 + levels, S + 1) = Weight
```

```
                Sheets("Sub-Net Values").Cells(8 + levels, S + 1) = Mon
                Sheets("Sub-Net Values").Cells(9 + levels, S + 1) = Rho
                Sheets("Sub-Net Values").Cells(10 + levels, S + 1) = Weight
                Sheets("Sensitivity Values").Cells(8 + levels, S + 1) = Mon
                Sheets("Sensitivity Values").Cells(9 + levels, S + 1) = Rho
                Sheets("Sensitivity Values").Cells(10 + levels, S + 1) = Weight
              Else
                For Q = 1 To P
                  Value = Application.InputBox("Enter value " & Q & " in level " & LevelA + 2 & " under
value " & O & " in level " & LevelA + 1 & " under value " & M & " in level " & LevelA & " under value "
& K & " in level " & I, "Value Hierarchy", Type:=2)
                  Weight = Application.InputBox("Enter weight " & Q & " in level " & LevelA + 2 & "
under value " & O & " in level " & LevelA + 1 & " under value " & M & " in level " & LevelA & " under
value " & K & " in level " & I, "Value Hierarchy", Type:=1)
                  Sheets("Values").Cells(8, S).Value = Value
                  Sheets("Values").Cells(8, S + 1).Value = Weight
                  Sheets("Sub-Net Values").Cells(8, S).Value = Value
                  Sheets("Sub-Net Values").Cells(8, S + 1).Value = Weight
                  Sheets("Sensitivity Values").Cells(8, S).Value = Value
                  Sheets("Sensitivity Values").Cells(8, S + 1).Value = Weight
                  R = InputBox("Enter number of values in level " & LevelA + 3 & " under value " & Q &
" in level " & LevelA + 2 & " under value " & O & " in level " & LevelA + 1 & " under value " & M & " in
level " & LevelA & " under value " & K & " in level " & I)
                  If R = 0 Then
                    Sheets("Input").Cells(1, Z).Value = Value
                    Sheets("Sub-Net Input").Cells(1, Z).Value = Value
                    Sheets("Scores").Cells(1, Z).Value = Value
                    Z = Z + 1
                    Sheets("Values").Cells(6 + levels, S) = "Low:"
                    Sheets("Values").Cells(7 + levels, S) = "High:"
                    Sheets("Values").Cells(8 + levels, S) = "Mono:"
                    Sheets("Values").Cells(9 + levels, S) = "Rho:"
                    Sheets("Values").Cells(10 + levels, S) = "Weight:"
                    Sheets("Sub-Net Values").Cells(6 + levels, S) = "Low:"
                    Sheets("Sub-Net Values").Cells(7 + levels, S) = "High:"
                    Sheets("Sub-Net Values").Cells(8 + levels, S) = "Mono:"
                    Sheets("Sub-Net Values").Cells(9 + levels, S) = "Rho:"
                    Sheets("Sub-Net Values").Cells(10 + levels, S) = "Weight:"
                    Sheets("Sensitivity Values").Cells(6 + levels, S) = "Low:"
                    Sheets("Sensitivity Values").Cells(7 + levels, S) = "High:"
                    Sheets("Sensitivity Values").Cells(8 + levels, S) = "Mono:"
                    Sheets("Sensitivity Values").Cells(9 + levels, S) = "Rho:"
                    Sheets("Sensitivity Values").Cells(10 + levels, S) = "Weight:"
                    Mon = Application.InputBox("Enter the monotonicity: Increasing or Decreasing",
Type:=2)
                    Rho = InputBox("Enter Rho")
                    Sheets("Values").Cells(8 + levels, S + 1) = Mon
                    Sheets("Values").Cells(9 + levels, S + 1) = Rho
                    Sheets("Values").Cells(10 + levels, S + 1) = Weight
                    Sheets("Sub-Net Values").Cells(8 + levels, S + 1) = Mon
                    Sheets("Sub-Net Values").Cells(9 + levels, S + 1) = Rho
                    Sheets("Sub-Net Values").Cells(10 + levels, S + 1) = Weight
                    Sheets("Sensitivity Values").Cells(8 + levels, S + 1) = Mon
                    Sheets("Sensitivity Values").Cells(9 + levels, S + 1) = Rho
```

```
                Sheets("Sensitivity Values").Cells(10 + levels, S + 1) = Weight
              Else
               For T = 1 To R
                  Value = Application.InputBox("Enter value " & R & " in level " & LevelA + 3 & "
under value " & Q & " in level " & LevelA + 2 & " under value " & O & " in level " & LevelA + 1 & "
under value " & M & " in level " & LevelA & " under value " & K & " in level " & I, "Value Hierarchy",
Type:=2)
                  Weight = Application.InputBox("Enter weight " & R & " in level " & LevelA + 3 &
" under value " & Q & " in level " & LevelA + 2 & " under value " & O & " in level " & LevelA + 1 & "
under value " & M & " in level " & LevelA & " under value " & K & " in level " & I, "Value Hierarchy",
Type:=1)
                  Sheets("Values").Cells(9, S).Value = Value
                  Sheets("Values").Cells(9, S + 1).Value = Weight
                  Sheets("Values").Cells(6 + levels, S) = "Low:"
                  Sheets("Values").Cells(7 + levels, S) = "High:"
                  Sheets("Values").Cells(8 + levels, S) = "Mono:"
                  Sheets("Values").Cells(9 + levels, S) = "Rho:"
                  Sheets("Values").Cells(10 + levels, S) = "Weight:"
                  Sheets("Sub-Net Values").Cells(9, S).Value = Value
                  Sheets("Sub-Net Values").Cells(9, S + 1).Value = Weight
                  Sheets("Sub-Net Values").Cells(6 + levels, S) = "Low:"
                  Sheets("Sub-Net Values").Cells(7 + levels, S) = "High:"
                  Sheets("Sub-Net Values").Cells(8 + levels, S) = "Mono:"
                  Sheets("Sub-Net Values").Cells(9 + levels, S) = "Rho:"
                  Sheets("Sub-Net Values").Cells(10 + levels, S) = "Weight:"
                  Sheets("Sensitivity Values").Cells(9, S).Value = Value
                  Sheets("Sensitivity Values").Cells(9, S + 1).Value = Weight
                  Sheets("Sensitivity Values").Cells(6 + levels, S) = "Low:"
                  Sheets("Sensitivity Values").Cells(7 + levels, S) = "High:"
                  Sheets("Sensitivity Values").Cells(8 + levels, S) = "Mono:"
                  Sheets("Sensitivity Values").Cells(9 + levels, S) = "Rho:"
                  Sheets("Sensitivity Values").Cells(10 + levels, S) = "Weight:"
                  Sheets("Input").Cells(1, Z).Value = Value
                  Sheets("Sub-Net Input").Cells(1, Z).Value = Value
                  Sheets("Scores").Cells(1, Z).Value = Value
                  Z = Z + 1
                  Mon = Application.InputBox("Enter the monotonicity: Increasing or Decreasing",
Type:=2)
                  Rho = InputBox("Enter Rho")
                  Sheets("Values").Cells(8 + levels, S + 1) = Mon
                  Sheets("Values").Cells(9 + levels, S + 1) = Rho
                  Sheets("Values").Cells(10 + levels, S + 1) = Weight
                  Sheets("Sub-Net Values").Cells(8 + levels, S + 1) = Mon
                  Sheets("Sub-Net Values").Cells(9 + levels, S + 1) = Rho
                  Sheets("Sub-Net Values").Cells(10 + levels, S + 1) = Weight
                  Sheets("Sensitivity Values").Cells(8 + levels, S + 1) = Mon
                  Sheets("Sensitivity Values").Cells(9 + levels, S + 1) = Rho
                  Sheets("Sensitivity Values").Cells(10 + levels, S + 1) = Weight
               Next T
              End If
              If Q <> P Then
              S = S + 2
              End If
            Next Q
```

```
                End If
                If O <> N Then
                S = S + 2
                End If
            Next O
        End If
        If M <> L Then
        S = S + 2
        End If
    Next M
  End If
  If K <> J Then
  S = S + 3
  End If
Next K

End Sub
```

## Scoring Module

```
'This module was modified from Leinarts Network Disruption Modeling Tool [20]
'It has been modified to incorporate the added capability of a user input
'value model.  It calculates the ranges of the single dimensional value
'functions and determines the score of each vulnerability set.


Option Explicit
Option Base 1


Dim NumCutsets As Long, L As Long, K As Long, NumMeasures As Integer



Sub ValueRange(CutsetCount As Long)


Dim I As Long, J As Long, M As Long, ValueArray() As Single
Dim Value_counter As Single, temp As String, comp As String, Value_average As Single

ReDim ValueArray(1 To CutsetCount)
NumMeasures = Sheets("Input").Range("B:iv").CurrentRegion.Columns.Count - 1
L = 2

For K = 1 To NumMeasures
M = 2
J = 2

   For I = 1 To CutsetCount
      Do While (Sheets("CutSets").Cells(I, J) <> "")
           temp = Sheets("CutSets").Cells(I, J).Value
25:
           comp = Sheets("Input").Cells(M, 1).Value
           If temp <> comp Then
              M = M + 1
              GoTo 25
           End If
           Value_counter = Value_counter + Sheets("Input").Cells(M, L)
           J = J + 1
           M = 2
      Loop
      Value_average = Value_counter / (J - 2)
      ValueArray(I) = Value_average
      With Sheets("Scores")
         .Cells(I + 1, 1) = "Vulnerability Set" & I
         .Cells(I + 1, L) = Application.Round(ValueArray(I), 2)
      End With
      Value_counter = 0
      J = 2
      M = 2
   Next I
   L = L + 1
```

```
Next K

End Sub


Sub RangesMaxMin(CutsetCount As Long)

Dim Min1 As Single, Max1 As Single, X As Long, Y As Long, Z As Long
Dim NumLevels As Integer, V As Long, W As String

Z = 66

NumLevels = Sheets("Stats").Cells(3, 10).Value
V = 1

For X = 1 To NumMeasures

45:
        W = Sheets("Values").Cells(4 + NumLevels + 2, V).Value
        If W = "" Then
           V = V + 1
           GoTo 45
        End If
        V = V + 1

     Min1 = Application.Min(Sheets("Scores").Range(Chr$(Z) + ":" + Chr$(Z)))
     Sheets("Values").Cells(4 + NumLevels + 2, V) = Application.Round(Min1, 2)
     Sheets("Sensitivity Values").Cells(4 + NumLevels + 2, V) = Application.Round(Min1, 2)
     Max1 = Application.Max(Sheets("Scores").Range(Chr$(Z) + ":" + Chr$(Z)))
     Sheets("Values").Cells(4 + NumLevels + 3, V) = Application.Round(Max1, 2)
     Sheets("Sensitivity Values").Cells(4 + NumLevels + 3, V) = Application.Round(Max1, 2)

     Z = Z + 1
     V = V + 1
Next X

End Sub

Sub Scores()
   NumCutsets = Sheets("CutSets").Range("a:iv").CurrentRegion.Rows.Count
   Sheets("Scores").Range("a2:iv16384").ClearContents
   Sheets("Scores").Activate
   Call ValueRange(NumCutsets)
   Call RangesMaxMin(NumCutsets)

End Sub
```

## Value Module

```
'This module was extensively modified from Lienart's Network Disruption
'Modeling Tool [20]. It incorporates the added capability of a user's inputed
'value model. This module calculates the value of each vulnerability set.

Option Explicit
Option Base 1


Sub Values()

Dim NumMeasures As Integer, NumLevels As Integer, Total As Double
Dim NumScores As Long, I As Long, J As Integer, K As Integer, Dummy As Variant
Dim temp As Integer, L As Integer, placer As Integer, C As Integer
Dim Msg, FirstRow As Integer, Counts As Integer, B As String, First As Variant
Dim Col As Integer, X As Integer, T As Integer, W As String, H As Double
Dim V As Integer, R As Integer, Z As Variant, S As Variant, Y As Double
Dim Counter As Integer, F As Double, E As Double, P As String, G As Integer


NumMeasures = Sheets("Input").Range("B:iv").CurrentRegion.Columns.Count - 1
NumScores = Sheets("CutSets").Range("a:iv").CurrentRegion.Rows.Count
NumLevels = Sheets("Stats").Cells(3, 10).Value
FirstRow = NumLevels + 14

Sheets("Values").Activate

B = Sheets("Values").Cells(FirstRow, 1).Value
If B = "" Then
   GoTo 30
End If
 G = 1
   For J = 1 To NumMeasures
10:
      W = Sheets("Values").Cells(FirstRow, G).Value
        If W = "" Then
           G = G + 1
           GoTo 10
        End If
      G = G + 1
   Next J
G = G + 2
15:
W = Sheets("Values").Cells(FirstRow, G).Value
   If W = "" Then
      GoTo 20:
   End If
   G = G + 1
   GoTo 15:
20:
Counts = G

With Sheets("Values")
```

```
   .Range(Cells(FirstRow - 1, 1), Cells(FirstRow + NumScores - 1, G - 1)).ClearContents
End With
Sheets("Sensitivity Values").Activate
With Sheets("Sensitivity Values")
   .Range(Cells(FirstRow - 1, 1), Cells(FirstRow + NumScores - 1, G - 1)).ClearContents
End With
Sheets("Values").Activate

30:
With Sheets("Values")

For I = 1 To NumScores
   .Cells(I + 13 + NumLevels, 1) = "Vulnerability Set " & I
   Sheets("Sensitivity Values").Cells(I + 13 + NumLevels, 1) = "Vulnerability Set " & I

   G = 1
   For J = 1 To NumMeasures

45:
      W = Sheets("Values").Cells(10 + NumLevels, G).Value
        If W = "" Then
           G = G + 1
           GoTo 45
        End If
      G = G + 1

      .Cells(I + 13 + NumLevels, G) = Application.Cells(10 + NumLevels, G).Value *
ValueE(Sheets("Scores").Cells(I + 1, J + 1).Value, .Cells(6 + NumLevels, G).Value, .Cells(7 +
NumLevels, G).Value, .Cells(8 + NumLevels, G).Value, .Cells(9 + NumLevels, G).Value)
      Sheets("Sensitivity Values").Cells(I + 13 + NumLevels, G) = Application.Cells(10 + NumLevels,
G).Value * ValueE(Sheets("Scores").Cells(I + 1, J + 1).Value, .Cells(6 + NumLevels, G).Value, .Cells(7 +
NumLevels, G).Value, .Cells(8 + NumLevels, G).Value, .Cells(9 + NumLevels, G).Value)
      G = G + 1
   Next J


   X = 1
   T = 1
   Col = G

100:
      W = Sheets("Values").Cells(4 + T, X).Value
      If X > G Then
         GoTo 300
      End If
      If W = "" Then
         X = X + 1
         GoTo 100
      End If


      For T = 2 To NumLevels
         W = Sheets("Values").Cells(4 + T, X).Value
```

```
        If W = "" Then
          T = T - 1
          GoTo 200
        End If
      Next T
      If T > NumLevels Then T = T - 1
200:
      V = X
      R = X + 1
      C = X + 2

202:
      Y = 0
      H = Sheets("Values").Cells(3 + T, R).Value
      P = Sheets("Values").Cells(3 + T, R - 1).Value
205:

        Z = Sheets("Values").Cells(3 + T, C).Value
        S = Sheets("Values").Cells(4 + T, R).Value
        Dummy = Sheets("Values").Cells(5 + T, R).Value
        If Z <> "" Or S Like "???*??" Or S = "" Then GoTo 210
        If Dummy = "" Then
          Y = (Sheets("Values").Cells(I + 13 + NumLevels, R).Value) + Y
         Else
          Y = (S * Sheets("Values").Cells(I + 13 + NumLevels, R).Value) + Y
        End If
        R = R + 2
        C = C + 1
      GoTo 205
210:
      If I = 1 Then
        Sheets("Values").Cells(I + 12 + NumLevels, Col) = P
        Sheets("Sensitivity Values").Cells(I + 12 + NumLevels, Col) = P
      End If
      Sheets("Values").Cells(I + 13 + NumLevels, Col) = Y * H
      Sheets("Sensitivity Values").Cells(I + 13 + NumLevels, Col) = Y * H
      Col = Col + 1
      If S Like "#*#" Then
        Counter = Counter + 1
        C = C + 2
        GoTo 202
      End If

220:
      If T - 2 > 0 Then
        T = T - 2
        Counter = Counter + 1
        If I = 1 Then
          Sheets("Values").Cells(I + 12 + NumLevels, Col) = Sheets("Values").Cells(4 + T, V).Value
          Sheets("Sensitivity Values").Cells(I + 12 + NumLevels, Col) = Sheets("Values").Cells(4 + T,
V).Value
        End If
        F = Sheets("Values").Cells(4 + T, V + 1).Value
        Do While Counter > 0
```

```
            E = (Sheets("Values").Cells(I + 13 + NumLevels, Col - Counter).Value + E)
            Counter = Counter - 1
          Loop
          Sheets("Values").Cells(I + 13 + NumLevels, Col) = F * E
          Sheets("Sensitivity Values").Cells(I + 13 + NumLevels, Col) = F * E
          Col = Col + 1
          GoTo 220
        End If

        Total = Sheets("Values").Cells(I + 13 + NumLevels, Col - 1) + Total

        X = X + 2
        T = 1
        E = 0

GoTo 100
300:
Counter = 0
If I = 1 Then
   Sheets("Values").Cells(I + 12 + NumLevels, Col).Value = "Total Score"
   Sheets("Sensitivity Values").Cells(I + 12 + NumLevels, Col).Value = "Total Score"
End If
Sheets("Values").Cells(I + 13 + NumLevels, Col).Value = Total
Sheets("Sensitivity Values").Cells(I + 13 + NumLevels, Col).Value = Total

Total = 0

Next I

End With

End Sub
```

## Value Function Module

```
'This module is taken directly from Leinart's Network Disruption
'Modeling Tool [20]. It is used in the value module routine and
'is used as the value function to put the evaluation measures
'on the same scale. The function is developed from Kirkwood [17].


Function ValueE(X, Low, High, Monotonicity, Rho)
  Select Case UCase(Monotonicity)
    Case "INCREASING"
      Difference = X - Low
    Case "DECREASING"
      Difference = High - X
  End Select
  If UCase(Rho) = "INFINITY" Then
    If (High - Low) = 0 Then
       ValueE = 0.5
    Else
    ValueE = Difference / (High - Low)
    End If
  Else
    ValueE = (1 - Exp(-Difference / Rho)) / (1 - Exp(-(High - Low) / Rho))
  End If
End Function
```

## Ranking Module

```
'This module was modified from Leinart's Network Disruption Modeling Tool [20].
'It incorporates the added capability of a user's inputed value model.
'It also adds the feature of identifying the components of the final list
'of vulnerability sets.  This module ranks the vulnerability sets on the
'value sheets by their values and also gets input from the users concerning
'how many sets to include in the final list of candidate vulnerability sets.

Dim Response1

Sub Ranking()

Dim FirstRow As Integer
Dim NumMeasures As Integer, M As Long
Dim NumScores As Long, I As Long, RowPlace As Long
Dim A, numrank As Integer, Counts As Integer
Dim Msg, W As String, G As Integer

NumMeasures = Sheets("Input").Range("B:iv").CurrentRegion.Columns.Count - 1
NumScores = Sheets("CutSets").Range("a:iv").CurrentRegion.Rows.Count
FirstRow = Sheets("Stats").Cells(3, 10).Value + 14
RowPlace = FirstRow
Sheets("TNL-Draft").Range("a:iv").ClearContents
Sheets("TNL-Final").Range("a:iv").ClearContents
Sheets("TNL-Draft").Activate

  G = 1
  For J = 1 To NumMeasures
45:
    W = Sheets("Values").Cells(FirstRow, G).Value
      If W = "" Then
        G = G + 1
        GoTo 45
      End If
    G = G + 1
  Next J
G = G + 2
50:
W = Sheets("Values").Cells(FirstRow, G).Value
  If W = "" Then
    GoTo 55:
  End If
  G = G + 1
  GoTo 50:
55:
Counts = G

For I = 1 To NumScores
  Sheets("TNL-Draft").Cells(I, 1) = Sheets("Values").Cells(RowPlace, 1)
  Sheets("TNL-Draft").Cells(I, 2) = Sheets("Values").Cells(RowPlace, Counts - 1)
  RowPlace = RowPlace + 1
Next I
```

```
dialrank:
  Message = "Enter the number of ranked vulnerability sets desired"
  Title = "Ranking"
  A = InputBox(Message, Title)

  If A = vbCancel Then
    Exit Sub
  End If

  If A = "" Then
    GoTo dialrank
  End If

  If A > NumScores Then
    Msg = "There are not that many existing vulnerability sets"
    MsgBox Msg
    GoTo dialrank
  End If

  Worksheets("TNL-Draft").Range("a:iv").CurrentRegion.Sort _
  key1:=Worksheets("TNL-Draft").Columns("B"), order1:=xlDescending

  Sheets("TNL-Final").Activate
  For I = 1 To A
    Sheets("TNL-Final").Cells(I, 2) = Sheets("TNL-Draft").Cells(I, 1)
    Sheets("TNL-Final").Cells(I, 1) = Application.Round(Sheets("TNL-Draft").Cells(I, 2), 4)
  Next I

  M = 1

  For I = 1 To A
    temp = Sheets("TNL-Final").Cells(I, 2).Value
25:
    comp = Sheets("CutSets").Cells(M, 1).Value
    If temp <> comp Then
      M = M + 1
      GoTo 25
    End If
    J = 2
    K = 3
30:
    If Sheets("CutSets").Cells(M, J) <> "" Then
      Sheets("TNL-Final").Cells(I, K).Value = Sheets("CutSets").Cells(M, J)
      J = J + 1
      K = K + 1
      GoTo 30
    End If
    M = 1
  Next I

End Sub
```

## Sensitivity and Persistency Analysis Module

```
'This module performs the sensitivity analysis and persistency analysis.


Sub SensitivityAnalysis()

Dim K As Double, I As Integer, J As Long, L As String, M As String, N As Integer, term As String
Dim O As Integer, P As Double, Q As String, R As String, S As Integer, Number As Integer
Dim iAnswer4 As String, B As Integer, Counter As Long, NumMeasures As Integer, Name As String
Dim Data As String, T As Long, D As Double, E As Double, F As Double, G As Integer, H As Integer
Dim iAnswer As String, iAnswer2 As Integer, iAnswer3 As Integer, Base As Double, A As String, Msg
Dim Z As Integer, Y As Integer, X As Integer, W As Variant, Ave As Double, Num As Double
Dim V As Integer, AA As Integer, U As Integer, BB As Integer, FF As Integer, CC As Integer
Dim EE As Integer, Data1 As String, Data2 As Integer, Data3 As Integer, Data4 As String
Dim GG As Integer, iAnswer5 As String, Data5 As String, NumLevels As Integer

Sheets("Stats").Activate
   With Sheets("Stats")
      Range("A8:D10").ClearContents
   End With

Sheets("Persistency Analysis").Activate
   With Sheets("Persistency Analysis")
      Range("A:Z").ClearContents
   End With

Sheets("Sensitivity Analysis").Activate
   With Sheets("Sensitivity Analysis")
      Range("A:Z").ClearContents
   End With

NumLevels = Sheets("Stats").Cells(3, 10).Value

5:
FF = InputBox("How many values would you like to perform one-way sensitivity analysis on?  Note: You
Can choose up to three at one time.")

If FF = False Then Exit Sub

If FF > 3 Then
   MsgBox ("Cannot do that many try again")
   GoTo 5
End If

For EE = 1 To FF

10:

iAnswer = InputBox("What is value " & EE & " that you would like to perform sensitivity analysis on?")
   If iAnswer = "" Then Exit Sub
   Sheets("Stats").Cells(EE + 7, 1).Value = iAnswer
iAnswer2 = InputBox("What level in the hierarchy is that value on?")
   If iAnswer2 = False Then Exit Sub
```

```
   Sheets("Stats").Cells(EE + 7, 2).Value = iAnswer2
iAnswer3 = InputBox("How many values are on that sublevel?")
   If iAnswer3 = False Then Exit Sub
   Sheets("Stats").Cells(EE + 7, 3).Value = iAnswer3

If iAnswer2 <> 1 Then
   iAnswer4 = InputBox("Which value is " & iAnswer & " under?")
      If iAnswer4 = False Then Exit Sub
      Sheets("Stats").Cells(EE + 7, 4).Value = iAnswer4
15:
   iAnswer5 = InputBox("Is " & iAnswer & " in the last level under " & iAnswer4 & "? (Type Y for yes or
N for no)")
      If iAnswer5 = False Then Exit Sub
      If iAnswer5 <> "Y" Then
         If iAnswer5 <> "N" Then
            MsgBox ("Please type Y for yes or N for no.")
            GoTo 15
         End If
      End If
      Sheets("Stats").Cells(EE + 7, 5).Value = iAnswer5
End If

Next EE

Number = InputBox("How many vulnerability sets would you like to include in the analysis?")
   If Number = False Then Exit Sub

MsgBox ("Note: The analysis will take the top " & Number & " vulnerability sets to perform the analysis,
ranging the weights of of the value between 0 and 1 at an increment of .1.")

CC = 3
GG = 3

For EE = 1 To FF

K = 0

Data1 = Sheets("Stats").Cells(EE + 7, 1).Value
Data2 = Sheets("Stats").Cells(EE + 7, 2).Value
Data3 = Sheets("Stats").Cells(EE + 7, 3).Value
Data4 = Sheets("Stats").Cells(EE + 7, 4).Value
Data5 = Sheets("Stats").Cells(EE + 7, 5).Value

   Sheets("Sensitivity Analysis").Cells(CC, 1) = Data1
   Sheets("Persistency Analysis").Cells(GG, 1) = Data1
   Sheets("Persistency Analysis").Cells(GG, 14) = Data1


   For J = 2 To 12
      Sheets("Sensitivity Analysis").Cells(CC, J) = K
      Sheets("Sensitivity Analysis").Cells(CC, J + 12) = K
      Sheets("Persistency Analysis").Cells(GG, J) = K
      Sheets("Persistency Analysis").Cells(GG, J + 13) = K
      K = K + 0.1
```

```
   Next J


   Sheets("Sensitivity Analysis").Cells(2, J) = "Base Value"

For I = 1 To Number
   Sheets("Sensitivity Analysis").Cells(I + CC, 1).Value = Sheets("TNL-Draft").Cells(I, 1).Value
   Sheets("Sensitivity Analysis").Cells(I + CC, 13).Value = Sheets("TNL-Draft").Cells(I, 2).Value
   Sheets("Sensitivity Analysis").Cells(I + CC + Number, 1).Value = Sheets("TNL-Draft").Cells(I,
1).Value & " Below 20"
Next I


N = 5
L = ""
O = 1

'Find Value

25:
Do While N < 10
   L = Sheets("Sensitivity Values").Cells(N, O).Value
   If L = Data1 Then
      O = O + 1
      GoTo 40
   End If
   N = N + 1
Loop
   If O = 26 Then
      Msg = MsgBox("The value you selected does not exist please try again", vbOKCancel)
      If Msg = vbCancel Then
         Exit Sub
      End If
      GoTo 10
   End If
O = O + 1
N = 5
GoTo 25

40:
Z = Data2 + 4

X = 2

If Data2 <> 1 Then
45:
   W = Sheets("Values").Cells(Z - 1, X - 1).Value
   If W <> Data4 Then
      X = X + 1
      GoTo 45
   End If
End If
```

```
S = 2
P = 0
Do While P < 1 Or P = 1

B = X

Base = Sheets("Values").Cells(N, O).Value
Sheets("Sensitivity Analysis").Cells(CC, 13).Value = Base

'Change Weights

For I = 1 To Data3
   term = Sheets("Values").Cells(Z, B - 1).Value
   If Data1 <> term Then
      Num = Sheets("Values").Cells(Z, B).Value
      Sheets("Sensitivity Values").Cells(Z, B).Value = Application.Round(((Num / (1 - Base)) * (1 - P)), 2)
      If Data5 = "Y" Then
         Sheets("Sensitivity Values").Cells(10 + NumLevels, B) = Application.Round(((Num / (1 - Base)) *
(1 - P)), 2)
      End If
   End If
   If Data1 = term Then
      Sheets("Sensitivity Values").Cells(Z, B).Value = P
      If Data5 = "Y" Then
         Sheets("Sensitivity Values").Cells(10 + NumLevels, B) = P
      End If
   End If
   If I <> Data3 Then
48:
      W = Sheets("Sensitivity Values").Cells(N, B + 1).Value
      If W = "" Then
         B = B + 1
         GoTo 48
      End If
      B = B + 2
     Else
      B = B + 2
   End If
Next I


   Sheets("Sensitivity Values").Cells(N, O).Value = P

'Recalculate Vulnerability Values
   Call SensitivityValues

'Rank Vulnerability Sets
   Call SensitivityTNL

'Record Vulnerability values and rank #
   For I = 1 To Number
   J = 1
50:
```

```
      Q = Sheets("Sensitivity Analysis").Cells(I + CC, 1).Value
      R = Sheets("Sensitivity TNL").Cells(J, 1).Value
      If Q Like R Then
         GoTo 60
      End If
      J = J + 1
      GoTo 50
60:
      Sheets("Sensitivity Analysis").Cells(I + CC, S).Value = Sheets("Sensitivity TNL").Cells(J, 2).Value
      Sheets("Sensitivity Analysis").Cells(I + CC, S + 12).Value = J
   Next I

G = 2
H = 14
T = 1
For I = 1 To Number
   Do
      T = Sheets("Sensitivity Analysis").Cells(I + CC, H).Value
      If T = 0 Then GoTo 80
      If T > 20 Then
         D = Sheets("Sensitivity Analysis").Cells(CC, H).Value

70:
         If G < 13 Then
            F = Sheets("Sensitivity Analysis").Cells(CC, G).Value
            If F = D Then
               E = Sheets("Sensitivity Analysis").Cells(I + CC, G).Value
               GoTo 75
            End If
            G = G + 1
         GoTo 70
         End If

75:
      Sheets("Sensitivity Analysis").Cells(I + CC + Number, G).Value = E
      End If
      G = 2
      H = H + 1
   Loop

80:
   H = 14
   G = 2
   T = 1
Next I

'Persistency Analysis

Data = "Sensitivity TNL"
NumMeasures = Sheets("Input").Range("A:A").CurrentRegion.Rows.Count - 1

For I = 1 To NumMeasures
   Name = Sheets("Input").Cells(I + 1, 1).Value
   With Worksheets(Data).Range("A1:Z20")
```

```
    Set C = .Find(What:=Name, LookIn:=xlValues)
    If Not C Is Nothing Then
      firstAddress = C.Address

      Do
        Counter = Counter + 1
        Set C = .FindNext(C)

      Loop While Not C Is Nothing And C.Address <> firstAddress
    End If
    End With

    Sheets("Persistency Analysis").Cells(I + GG, 1).Value = Name
    Sheets("Persistency Analysis").Cells(I + GG, S).Value = Counter
    Counter = 0
Next I

T = 1
V = GG + 1
For I = 1 To NumMeasures
    For J = 2 To 12
      U = Sheets("Persistency Analysis").Cells(I + GG, J).Value
      If U <> 0 Then
        For AA = 14 To 25
          Sheets("Persistency Analysis").Cells(V, AA).Value = Sheets("Persistency Analysis").Cells(I +
GG, T).Value
          T = T + 1
        Next AA
        V = V + 1
        GoTo 90
      End If
      T = 1
    Next J
90:
    T = 1
Next I

    P = P + 0.1
    S = S + 1

Loop

'Put Back to Base Value

P = Base

B = X

Base = Sheets("Values").Cells(N, O).Value
Sheets("Sensitivity Analysis").Cells(CC, 13).Value = Base

For I = 1 To Data3
    term = Sheets("Values").Cells(Z, B - 1).Value
    If Data1 <> term Then
```

```
      Num = Sheets("Values").Cells(Z, B).Value
      Sheets("Sensitivity Values").Cells(Z, B).Value = Application.Round(((Num / (1 - Base)) * (1 - P)), 2)
      If Data5 = "Y" Then
         Sheets("Sensitivity Values").Cells(10 + NumLevels, B).Value = Application.Round(((Num / (1 -
Base)) * (1 - P)), 2)
      End If
   End If
   If Data1 = term Then
      Sheets("Sensitivity Values").Cells(Z, B).Value = P
      If Data5 = "Y" Then
         Sheets("Sensitivity Values").Cells(10 + NumLevels, B).Value = P
      End If
   End If
   If I <> Data3 Then
100:
      W = Sheets("Sensitivity Values").Cells(N, B + 1).Value
      If W = "" Then
         B = B + 1
         GoTo 100
      End If
      B = B + 2
     Else
      B = B + 2
   End If
Next I

Sheets("Sensitivity Values").Cells(N, O).Value = P

Call ChartMaker(Data1, Number, CC)
Call Chart(Data1, V, GG)

CC = CC + (2 * Number) + 2
GG = GG + NumMeasures + 2
Next EE


End Sub

Sub SensitivityValues()

Dim NumMeasures As Integer, NumLevels As Integer, Total As Double
Dim NumScores As Long, I As Long, J As Long, K As Integer
Dim temp As Integer, L As Integer, placer As Integer, C As Integer
Dim Msg, FirstRow As Integer, A As Integer, Dummy As Variant
Dim Col As Integer, X As Integer, T As Long, W As String, H As Double
Dim V As Integer, R As Integer, Z As Variant, S As Variant, Y As Double
Dim Counter As Integer, F As Double, E As Double, P As String, G As Integer


NumMeasures = Sheets("Input").Range("B:iv").CurrentRegion.Columns.Count - 1
NumScores = Sheets("CutSets").Range("a:iv").CurrentRegion.Rows.Count
NumLevels = Sheets("Stats").Cells(3, 10).Value
FirstRow = NumLevels + 14
```

```
G = 1
  For J = 1 To NumMeasures
10:
    W = Sheets("Sensitivity Values").Cells(FirstRow, G).Value
      If W = "" Then
        G = G + 1
        GoTo 10
      End If
    G = G + 1
  Next J
G = G + 2
A = G
15:
W = Sheets("Sensitivity Values").Cells(FirstRow, A).Value
  If W = "" Then
    GoTo 20:
  End If
  A = A + 1
  GoTo 15:
20:

Sheets("Sensitivity Values").Activate
  With Sheets("Sensitivity Values")
    .Range(Cells(FirstRow - 1, 1), Cells(FirstRow + NumScores - 1, A)).ClearContents
  End With


30:
With Sheets("Sensitivity Values")

For I = 1 To NumScores
  .Cells(I + 13 + NumLevels, 1) = "Vulnerability Set " & I

  G = 1
  For J = 1 To NumMeasures

45:
    W = Sheets("Sensitivity Values").Cells(10 + NumLevels, G).Value
      If W = "" Then
        G = G + 1
        GoTo 45
      End If
    G = G + 1

    .Cells(I + 13 + NumLevels, G) = Application.Cells(10 + NumLevels, G).Value *
ValueE(Sheets("Scores").Cells(I + 1, J + 1).Value, .Cells(6 + NumLevels, G).Value, .Cells(7 +
NumLevels, G).Value, .Cells(8 + NumLevels, G).Value, .Cells(9 + NumLevels, G).Value)

    G = G + 1

  Next J

  X = 1
```

```
  T = 1
  Col = G

100:
    W = Sheets("Sensitivity Values").Cells(4 + T, X).Value
    If X > G Then
      GoTo 300
    End If
    If W = "" Then
      X = X + 1
      GoTo 100
    End If


    For T = 2 To NumLevels
      W = Sheets("Sensitivity Values").Cells(4 + T, X).Value

      If W = "" Then
        T = T - 1
        GoTo 200
      End If
    Next T
    If T > NumLevels Then T = T - 1
200:
    V = X
    R = X + 1
    C = X + 2

202:
    Y = 0
    H = Sheets("Sensitivity Values").Cells(3 + T, R).Value
    P = Sheets("Sensitivity Values").Cells(3 + T, R - 1).Value
205:

      Z = Sheets("Sensitivity Values").Cells(3 + T, C).Value
      S = Sheets("Sensitivity Values").Cells(4 + T, R).Value
      Dummy = Sheets("Sensitivity Values").Cells(5 + T, R).Value
      If Z <> "" Or S Like "???*??" Or S = "" Then GoTo 210
      If Dummy = "" Then
        Y = Sheets("Sensitivity Values").Cells(I + 13 + NumLevels, R).Value + Y
       Else
        Y = (S * Sheets("Sensitivity Values").Cells(I + 13 + NumLevels, R).Value) + Y
      End If
      R = R + 2
      C = C + 1
    GoTo 205
210:
    If I = 1 Then
      Sheets("Sensitivity Values").Cells(I + 12 + NumLevels, Col) = P
    End If
    Sheets("Sensitivity Values").Cells(I + 13 + NumLevels, Col) = Y * H
    Col = Col + 1
    If S Like "#*#" Then
      Counter = Counter + 1
```

```
        C = C + 2
        GoTo 202
    End If

220:
    If T - 2 > 0 Then
        T = T - 2
        Counter = Counter + 1
        If I = 1 Then
            Sheets("Sensitivity Values").Cells(I + 12 + NumLevels, Col) = Sheets("Values").Cells(4 + T,
V).Value
        End If
        F = Sheets("Sensitivity Values").Cells(4 + T, V + 1).Value
        Do While Counter > 0
            E = (Sheets("Sensitivity Values").Cells(I + 13 + NumLevels, Col - Counter).Value + E)
            Counter = Counter - 1
        Loop
        Sheets("Sensitivity Values").Cells(I + 13 + NumLevels, Col) = F * E
        Col = Col + 1
        GoTo 220
    End If

    Total = Sheets("Sensitivity Values").Cells(I + 13 + NumLevels, Col - 1) + Total

    X = X + 2
    T = 1
    E = 0

GoTo 100
300:
Counter = 0
If I = 1 Then
   Sheets("Sensitivity Values").Cells(I + 12 + NumLevels, Col).Value = "Total Score"
End If
Sheets("Sensitivity Values").Cells(I + 13 + NumLevels, Col).Value = Total

Total = 0

Next I

End With
Sheets("Sensitivity Analysis").Activate

End Sub

Sub SensitivityTNL()

Dim FirstRow As Integer
Dim NumMeasures As Integer
Dim NumScores As Long, I As Long, RowPlace As Long
Dim A, numrank As Integer
Dim Msg, M As Long

NumMeasures = Sheets("Input").Range("B:iv").CurrentRegion.Columns.Count - 1
```

```
NumScores = Sheets("CutSets").Range("A:Z").CurrentRegion.Rows.Count
FirstRow = Sheets("Stats").Cells(3, 10).Value + 14
RowPlace = FirstRow
Sheets("Sensitivity TNL").Range("A:Z").ClearContents

 G = 1
   For J = 1 To NumMeasures
45:
      W = Sheets("Sensitivity Values").Cells(FirstRow, G).Value
        If W = "" Then
           G = G + 1
           GoTo 45
        End If
      G = G + 1
   Next J
G = G + 2
50:
W = Sheets("Sensitivity Values").Cells(FirstRow, G).Value
   If W = "" Then
      GoTo 55:
   End If
   G = G + 1
   GoTo 50:
55:
Counts = G

For I = 1 To NumScores
   Sheets("Sensitivity TNL").Cells(I, 1) = Sheets("Sensitivity Values").Cells(RowPlace, 1)
   Sheets("Sensitivity TNL").Cells(I, 2) = Sheets("Sensitivity Values").Cells(RowPlace, Counts - 1)
   RowPlace = RowPlace + 1
Next I


Worksheets("Sensitivity TNL").Range("A:iv").CurrentRegion.Sort _
key1:=Worksheets("Sensitivity TNL").Columns("B"), order1:=xlDescending

M = 1

   For I = 1 To 20
      temp = Sheets("Sensitivity TNL").Cells(I, 1).Value
25:
      comp = Sheets("CutSets").Cells(M, 1).Value
      If temp <> comp Then
         M = M + 1
         GoTo 25
      End If
      J = 2
      K = 3
30:
      If Sheets("CutSets").Cells(M, J) <> "" Then
         Sheets("Sensitivity TNL").Cells(I, K).Value = Sheets("CutSets").Cells(M, J)
         J = J + 1
         K = K + 1
         GoTo 30
```

```
    End If
    M = 1
  Next I


End Sub

Sub ChartMaker(Val As String, Num As Integer, DD As Integer)

Range(Cells(DD, 1), Cells(2 * Num + DD, 12)).Name = "TestRange"


Charts.Add

ActiveChart.ChartWizard _
   Source:=Sheets("Sensitivity Analysis").Range("TestRange"), _
   Gallery:=xlLine, _
   Format:=4, _
   PlotBy:=xlRows, _
   CategoryLabels:=1, _
   SeriesLabels:=1, _
   HasLegend:=True, Title:="One-Way Sensitivity Analysis", CategoryTitle:=Val & " Weight",
ValueTitle:="Vulnerability Set Value"

ActiveChart.Location xlLocationAutomatic, "Sensitivity Chart"

End Sub


Sub Chart(Val As String, Num As Integer, DD As Integer)

Sheets("Persistency Analysis").Activate

Range(Cells(DD, 14), Cells(Num - 1, 25)).Name = "TestRange"

Charts.Add

ActiveChart.ChartWizard _
   Source:=Sheets("Persistency Analysis").Range("TestRange"), _
   Gallery:=xlLine, _
   Format:=1, _
   PlotBy:=xlRows, _
   CategoryLabels:=1, _
   SeriesLabels:=1, _
   HasLegend:=True, Title:="Persistency Analysis", CategoryTitle:=Val & "Weight",
ValueTitle:="Recurrence"

ActiveChart.Location Where:=xlLocationAutomatic, _
   Name:="Persistency Chart"

End Sub
```

## Occurrence Analysis Module

```
'This module is used to perform Occurrence Analysis which determines
'the number of times a node or link appears in the list of vulnerability
'sets.  It then calculates the percent of occurrence and plots it on a chart.


Sub Occurrence()

Dim Counter As Long, NumMeasures As Integer, I As Integer
Dim Name As String, Data As String, NumCuts As Long

Sheets("Occurrence Analysis").Activate
   With Sheets("Occurrence Analysis")
      Range("A:C").ClearContents
   End With

10:
Data = "CutSets"

NumCuts = Sheets("CutSets").Range("A:A").CurrentRegion.Rows.Count
NumMeasures = Sheets("Input").Range("A:A").CurrentRegion.Rows.Count - 1

For I = 1 To NumMeasures

   Name = Sheets("Input").Cells(I + 1, 1).Value

   With Worksheets(Data).Range("A:Z")

   Set C = .Find(What:=Name, LookIn:=xlValues)

   If Not C Is Nothing Then
      firstAddress = C.Address

      Do
         Counter = Counter + 1
         Set C = .FindNext(C)
      Loop While Not C Is Nothing And C.Address <> firstAddress

   End If

   End With
   Sheets("Occurrence Analysis").Cells(I, 1).Value = Name
   Sheets("Occurrence Analysis").Cells(I, 3).Value = Counter
   Sheets("Occurrence Analysis").Cells(I, 2).Value = Application.Round((Counter / NumCuts), 2)

   Counter = 0

Next I

Call Chart

End Sub
```

```
Sub Chart()

Num = Sheets("Occurrence Analysis").Range("A:A").Rows.Count

Range(Cells(1, 1), Cells(Num, 2)).Name = "TestRange"

Charts.Add

ActiveChart.ChartWizard _
   Source:=Sheets("Occurrence Analysis").Range("TestRange"), _
   Gallery:=xlColumn, _
   Format:=2, _
   PlotBy:=xlColumns, _
   CategoryLabels:=1, _
   SeriesLabels:=1, _
   HasLegend:=False, Title:="Occurrence Analysis", CategoryTitle:="Component", ValueTitle:="Percent
Occurrence"

ActiveChart.Location Where:=xlLocationAutomatic, _
   Name:="Occurrence Chart"

End Sub
```

## Sub-Network Input Module

```
'This module is used to input the data for the sub-network.
'Components can either be manually input or the routine will
'let the user input the sub-network components for the nodes
'which are found in the final list of vulnerability sets.

Dim I As Integer, K As Integer, J As Integer, M As Integer, H As Integer, Number As Integer
Dim temp As String, comp As String, L As Variant, N As Variant, Data As String

Sub SubNetworkInput()

1:
iAnswer = MsgBox("Do you want to use the most critical nodes for the sub-network?", vbYesNo, "Sub-
Network")
If iAnswer = vbYes Then GoTo Continue
If iAnswer = vbNo Then
   MsgBox ("Please enter sub-network components on worksheet")
   Exit Sub
End If

Continue:

Sheets("Sub-Net Input").Range("A2:iv16834").ClearContents
Sheets("Sub-Net Input").Range("A2:iv16834").ClearFormats
Sheets("Sub-Net Input").Activate

I = Sheets("TNL-Final").Range("a:iv").CurrentRegion.Rows.Count
H = 2
K = 3
For J = 1 To I
10:
   Do While Sheets("TNL-Final").Cells(J, K) <> ""
      temp = Sheets("TNL-Final").Cells(J, K).Value

      Number = Sheets("Sub-Net Input").Range("A:iv").CurrentRegion.Rows.Count
      For A = 1 To Number
         Data = Sheets("Sub-Net Input").Cells(A, 1)
         If temp = Data Then
            K = K + 1
            GoTo 10
         End If
      Next A

      comp = "???? #"
      If temp Like comp Then
         Sheets("Sub-Net Input").Cells(H, 1) = temp
         Sheets("Sub-Net Input").Cells(H, 1).Font.Color = RGB(255, 0, 0)
         iAnswer1 = MsgBox("Enter Sub-Network components for " & temp, vbOKCancel, "Sub-Network
Components")
         H = H + 1
         If iAnswer1 = vbCancel Then Exit Sub

         MsgBox ("When node entry is completed type End")
```

```
20:
        iAnswer2 = Application.InputBox("Enter Sub-Node", "Nodes", Type:=2)
        If iAnswer2 = "End" Or iAnswer2 = "end" Then GoTo 30
        If iAnswer2 = False Then Exit Sub
        Sheets("Sub-Net Input").Cells(H, 1) = iAnswer2
        H = H + 1
        GoTo 20

30:
        iAnswer3 = MsgBox("Click OK to enter Links. Click Cancel to exit routine.", vbOKCancel, "Sub-
Network " & temp)

        If iAnswer3 = vbCancel Then Exit Sub

        MsgBox ("When arc entry is completed enter End for the tail.")

40:
        N = Application.InputBox("Enter arc tail", "Link", Type:=2)

          If N = "End" Or N = "end" Then
            GoTo 50
          End If
          If N = False Then Exit Sub

          L = Application.InputBox("Enter arc head", "Link", Type:=2)

          If L = False Then Exit Sub

          Sheets("Sub-Net Input").Cells(H, 1) = "Link " & N & " to " & L
          H = H + 1


          GoTo 40
50:

      End If
      K = K + 1
    Loop
    K = 3
Next J

End Sub
```

## Sub-Network Values Module

```
'This module is modified from the Values module.  It is used to on the sub-network.
'It calculates the value of each vulnerability using the data on the sub-net input sheet.

Option Explicit
Option Base 1


Sub SubNetValues()

Dim NumMeasures As Integer, NumLevels As Integer, Total As Double
Dim NumScores As Integer, I As Integer, J As Integer, K As Integer
Dim temp As Integer, L As Integer, placer As Integer, C As Integer
Dim Msg, FirstRow As Integer, Counts As Integer, B As String, First As Variant
Dim Col As Integer, X As Integer, T As Integer, W As String, H As Double
Dim V As Integer, R As Integer, Z As Variant, S As Variant, Y As Double
Dim Counter As Integer, F As Double, E As Double, P As String, G As Integer
Dim Dummy As Variant

NumMeasures = Sheets("Sub-Net Input").Range("B:iv").CurrentRegion.Columns.Count - 1
NumScores = Sheets("Sub-Net Input").Range("a:iv").CurrentRegion.Rows.Count - 1
NumLevels = Sheets("Stats").Cells(3, 10).Value
FirstRow = NumLevels + 14

Sheets("Sub-Net Values").Activate

Call SubRangesMaxMin

B = Sheets("Sub-Net Values").Cells(FirstRow, 1).Value
If B = "" Then
   GoTo 30
End If
G = 1
   For J = 1 To NumMeasures
10:
      W = Sheets("Sub-Net Values").Cells(FirstRow, G).Value
        If W = "" Then
           G = G + 1
           GoTo 10
        End If
      G = G + 1
   Next J
G = G + 2
15:
W = Sheets("Sub-Net Values").Cells(FirstRow, G).Value
   If W = "" Then
      GoTo 20:
   End If
   G = G + 1
   GoTo 15
20:
Counts = G
```

```
With Sheets("Sub-Net Values")
   .Range(Cells(FirstRow - 1, 1), Cells(FirstRow + NumScores - 1, G - 1)).ClearContents
End With

30:
With Sheets("Sub-Net Values")

For I = 1 To NumScores
   .Cells(I + 13 + NumLevels, 1) = "Vulnerability " & I

   G = 1
   For J = 1 To NumMeasures
45:
      W = Sheets("Sub-Net Values").Cells(10 + NumLevels, G).Value
         If W = "" Then
            G = G + 1
            GoTo 45
         End If
      G = G + 1

      .Cells(I + 13 + NumLevels, G) = Application.Cells(10 + NumLevels, G).Value *
ValueE(Sheets("Sub-Net Input").Cells(I + 1, J + 1).Value, .Cells(6 + NumLevels, G).Value, .Cells(7 +
NumLevels, G).Value, .Cells(8 + NumLevels, G).Value, .Cells(9 + NumLevels, G).Value)
      G = G + 1
   Next J

   X = 1
   T = 1
   Col = G

100:
   W = Sheets("Sub-Net Values").Cells(4 + T, X).Value
   If X > G Then
      GoTo 300
   End If
   If W = "" Then
      X = X + 1
      GoTo 100
   End If

   For T = 2 To NumLevels
      W = Sheets("Sub-Net Values").Cells(4 + T, X).Value

      If W = "" Then
         T = T - 1
         GoTo 200
      End If
   Next T
   If T > NumLevels Then T = T - 1
200:
   V = X
   R = X + 1
   C = X + 2
```

```
202:
  Y = 0
  H = Sheets("Sub-Net Values").Cells(3 + T, R).Value
  P = Sheets("Sub-Net Values").Cells(3 + T, R - 1).Value
205:
    Z = Sheets("Sub-Net Values").Cells(3 + T, C).Value
    S = Sheets("Sub-Net Values").Cells(4 + T, R).Value
    Dummy = Sheets("Sub-Net Values").Cells(5 + T, R).Value
    If Z <> "" Or S Like "???*??" Or S = "" Then GoTo 210
    If Dummy = "" Then
      Y = Sheets("Sub-Net Values").Cells(I + 13 + NumLevels, R).Value + Y
     Else
      Y = (S * Sheets("Sub-Net Values").Cells(I + 13 + NumLevels, R).Value) + Y
    End If
    R = R + 2
    C = C + 1
    GoTo 205
210:
    If I = 1 Then
      Sheets("Sub-Net Values").Cells(I + 12 + NumLevels, Col) = P
    End If

    Sheets("Sub-Net Values").Cells(I + 13 + NumLevels, Col) = Y * H
    Col = Col + 1
    If S Like "#*#" Then
      Counter = Counter + 1
      C = C + 2
      GoTo 202
    End If

220:

    If T - 2 > 0 Then
      T = T - 2
      Counter = Counter + 1
      If I = 1 Then
        Sheets("Sub-Net Values").Cells(I + 12 + NumLevels, Col) = Sheets("Sub-Net Values").Cells(4 +
T, V).Value
      End If
      F = Sheets("Sub-Net Values").Cells(4 + T, V + 1).Value
      Do While Counter > 0
        E = (Sheets("Sub-Net Values").Cells(I + 13 + NumLevels, Col - Counter).Value + E)
        Counter = Counter - 1
      Loop
      Sheets("Sub-Net Values").Cells(I + 13 + NumLevels, Col) = F * E
      Col = Col + 1
      GoTo 220
    End If

    Total = Sheets("Sub-Net Values").Cells(I + 13 + NumLevels, Col - 1) + Total

    X = X + 2
    T = 1
    E = 0
```

```
GoTo 100
300:
Counter = 0
If I = 1 Then
   Sheets("Sub-Net Values").Cells(I + 12 + NumLevels, Col).Value = "Total Score"
End If
Sheets("Sub-Net Values").Cells(I + 13 + NumLevels, Col).Value = Total

Total = 0

Next I

End With

End Sub


Sub SubRangesMaxMin()

Dim Min1 As Single, Max1 As Single, X As Integer, Y As Integer, Z As Integer
Dim NumMeasures As Integer, NumLevels As Integer, V As Integer, W As String

NumMeasures = Sheets("Sub-Net Input").Range("B:iv").CurrentRegion.Columns.Count - 1
NumLevels = Sheets("Stats").Cells(3, 10).Value

Z = 66
V = 1

For X = 1 To NumMeasures

45:
    W = Sheets("Sub-Net Values").Cells(4 + NumLevels + 2, V).Value
    If W = "" Then
       V = V + 1
       GoTo 45
    End If
    V = V + 1

    Min1 = Application.Min(Sheets("Sub-Net Input").Range(Chr$(Z) + ":" + Chr$(Z)))
    Sheets("Sub-Net Values").Cells(4 + NumLevels + 2, V) = Application.Round(Min1, 2)
    Max1 = Application.Max(Sheets("Sub-Net Input").Range(Chr$(Z) + ":" + Chr$(Z)))
    Sheets("Sub-Net Values").Cells(4 + NumLevels + 3, V) = Application.Round(Max1, 2)

    Z = Z + 1
    V = V + 1
Next X

End Sub
```

## Sub-Network Ranking Module

```
'This module was modified from the TNL Mod module.  It is modified to perform the
'functions for the sub-network. It ranks the vulnerabilities on the sub-net value
'sheet by their values and lists the sub-network components for each.


Sub SubNetRanking()

Dim FirstRow As Integer
Dim NumMeasures As Integer
Dim NumScores As Integer, I As Integer, RowPlace As Integer
Dim numrank As Integer
Dim Msg

NumMeasures = Sheets("Sub-Net Input").Range("B:iv").CurrentRegion.Columns.Count - 1
NumScores = Sheets("Sub-Net Input").Range("a:iv").CurrentRegion.Rows.Count - 1
FirstRow = Sheets("Stats").Cells(3, 10).Value + 14
RowPlace = FirstRow
Sheets("Sub-Net TNL").Range("a:iv").ClearContents
Sheets("Sub-Net TNL").Activate

G = 1
  For J = 1 To NumMeasures
45:
    W = Sheets("Sub-Net Values").Cells(FirstRow, G).Value
      If W = "" Then
        G = G + 1
        GoTo 45
      End If
    G = G + 1
  Next J
G = G + 2
50:
W = Sheets("Sub-Net Values").Cells(FirstRow, G).Value
  If W = "" Then
    GoTo 55:
  End If
  G = G + 1
  GoTo 50:
55:
Counts = G

For I = 1 To NumScores
  Sheets("Sub-Net TNL").Cells(I, 2) = Sheets("Sub-Net Values").Cells(RowPlace, 1)
  Sheets("Sub-Net TNL").Cells(I, 1) = Sheets("Sub-Net Values").Cells(RowPlace, Counts - 1)
  RowPlace = RowPlace + 1
Next I


  M = 1

  For I = 2 To NumScores + 1
```

```
    If Sheets("Sub-Net Input").Cells(I, 1).Value = "" Then
       comp = Sheets("Sub-Net Input").Cells(I, 2).Value
      Else
       comp = Sheets("Sub-Net Input").Cells(I, 1).Value
    End If

    Sheets("Sub-Net TNL").Cells(I - 1, 3).Value = comp

  Next I

 Worksheets("Sub-Net TNL").Range("a:iv").CurrentRegion.Sort _
  key1:=Worksheets("Sub-Net TNL").Columns("A"), order1:=xlDescending

End Sub
```

# Vita

Captain Karilynne Wallace was born on 11 August, 1972 in Gunnison, Utah. She graduated from Gunnison Valley High School in 1990 and entered undergraduate studies at Utah State University in Logan, Utah. She graduated with a Bachelor of Science degree in Mathematics in June 1994. She received her commission upon graduation from college and her completion of training with the Air Force Reserve Officer Training Corps.

Her first assignment was at Vandenberg Air Force Base, California as a Program Manager and Range Control Officer. In August 1997, she entered the Graduate School of Engineering, Air Force Institute of Technology. Captain Wallace's follow-on assignment was to the 17th Test Squadron at Schriever Air Force Base in Colorado.

Permanent Address: 198 East Order Road
Mayfield, UT 84643

| REPORT DOCUMENTATION PAGE | Form Approved OMB No. 0704-0188 |
|---|---|



| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | March 1999 | Master's Thesis |

**4. TITLE AND SUBTITLE**
MULTI-DISCIPLINE NETWORK VULNERABILITY ASSESSMENT

**5. FUNDING NUMBERS**

**6. AUTHOR(S)**
Karilynne Wallace, Captain, USAF

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Air Force Institute of Technology
2750 P Street
WPAFB OH 45433-7765

**8. PERFORMING ORGANIZATION REPORT NUMBER**
AFIT/GSO/ENS/99M-01

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
AFSPACECOM/DOIK
150 Vandenberg Street (Bldg 1)
Peterson AFB CO 80914-4150

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**
Advisor: Assistant Professor W. P. Murdock, Jr., Maj AFIT/ENS
(937) 255-6565 ext 4339 william.murdock@afit.af.mil

**12a. DISTRIBUTION AVAILABILITY STATEMENT**
Approved for public release, distribution unlimited.

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(Maximum 200 words)*

"Air Force Space Command has a mission to provide real-time, survivable, and enduring communications, surveillance, environmental monitoring, navigation, and warning..." according to Air Force Space Handbook - A War Fighters Guide to Space. In order to prevent disruption of the critical networks in Space Command, a vulnerability assessment must be performed. The set of components called a vulnerability set, which could cause network disruption, must be identified and prioritized. In order to move beyond subjective prioritization, implementation of a quantitative methodology that measures the value of each vulnerability set is needed.

This thesis proposes a technique for vulnerability assessment of a multi-disciplined network consisting of components that satisfy primary functions and secondary supporting functions. A Network Vulnerability Assessment Tool was developed and used to analyze the network. The tool models the network as nodes and links. A cut-set algorithm generates potential vulnerability sets. The overall objective is to find sets that cause critical network disruptions. Value-focused thinking and decision analysis techniques are used to rank the vulnerability sets according to decision maker preferences. The network vulnerability sets are ranked to provide an overall vulnerability assessment. Sensitivity analysis provides insight into the driving factors of the vulnerability set prioritization.

| 14. SUBJECT TERMS | 15. NUMBER OF PAGES |
|---|---|
| Network Disruption, Vulnerabilities, Graph Theory, Cut-Sets, Value-Focused Thinking, Decision Analysis | 173 |
| | 16. PRICE CODE |

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | UL |